VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

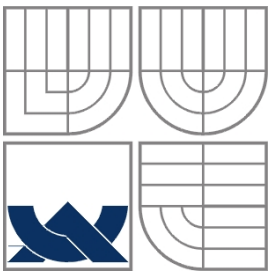
APLIKACE PRO PODPORU KALIBRACE PÍSTOVÝCH PIPET

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

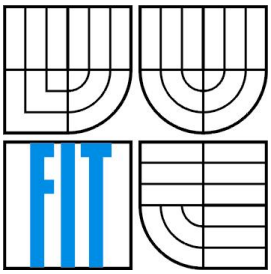
AUTOR PRÁCE
AUTHOR

BOHUMÍR VOSPĚL

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

APLIKACE PRO PODPORU KALIBRACE PÍSTOVÝCH PIPET

APPLICATION FOR SUPPORT OF CALIBRATION OF A PISTON PIPETTE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Bohumír Vospěl

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Filip Orság, Ph.D.

BRNO 2011

Abstrakt

Tato práce se zabývá částečným zautomatizováním pracovního postupu při kalibraci pístových pipet. Analyzuje možnosti využití stávajících měřidel laboratoře, zejména se pak zaměřuje na propojení těchto přístrojů s osobním počítačem a jejich vzdálenou obsluhu. Naměřené hodnoty umožní exportovat do sešitu aplikace Microsoft Excel. Navrhuje kompletní řešení této problematiky.

Abstract

This thesis occupies itself with the partial automation of the workflow for calibration of piston pipettes. It analyses the possibilities of the use of the current laboratory gauges, and focuses particularly on the connecting of these devices with a personal computer and their remote control. It also enables the exportation of the measured values into Microsoft Excel application and suggests a complete solution of these problems.

Klíčová slova

kalibrace, komunikační protokol, USB, RS232, Microsoft Excel, Modbus RTU

Keywords

calibration, communication protocol, USB, RS232, Microsoft Excel, Modbus RTU

Citace

Bohumír Vospěl: Aplikace pro podporu kalibrace pístových pipet, bakalářská práce, Brno, FIT VUT v Brně, 2011

Měření a zpracování hodnot z externích přístrojů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Filipa Orsága, Ph.D.
Další informace mi poskytl Ing. Jiří Pokorný a zadavatel.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Bohumír Vospěl
18. 5. 2011

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce Ing. Filipu Orságovi, Ph.D., Ing. Jiřímu Pokornému a Mgr. Martině Vičarové za poskytnutou pomoc, odborné vedení a čas věnovaný při konzultaci tématu mé bakalářské práce.

Obsah

1 Úvod

Odměřování přesného objemu tekutin je jednou z důležitých laboratorních činností při chemických, biochemických, mikrobiologických a klinických analýzách. Na přesnosti měření objemu významně závisí hodnota výsledků analýz. K odměřování malých objemů, na příklad biologických materiálů, se používají pístové pipety s jednorázovými špičkami. Dávkovací přístroje, na jejichž přesnosti závisí přesnost výsledků analýzy, je nutné pravidelně kalibrovat. Kalibrace se provádí gravimetrickou metodou, která je založena na stanovení objemu destilované vody vypuštěné z pístových pipet. Tento objem se určí z hmotnosti destilované vody a z hodnoty její hustoty. Postup kalibrace pístových pipet je popsán v normě [1] a výpočet v technické zprávě [2].

Kalibrační laboratoř Českého metrologického institutu využívá pro kalibraci pístových pipet přesné analytické váhy, odporový teploměr s digitální vyhodnocovací jednotkou, přesný tlakoměr a vlhkoměr. Vzhledem k velkému počtu kalibrací je nepraktické, aby laborant při každém měření musel odečítat hodnoty ručně z tohoto poměrně velkého počtu přístrojů a zaznamenávat tyto hodnoty do poznámek, které jsou následně ručně přepisovány do počítače. Proto vznikla myšlenka celý tento postup zefektivnit automatizovaným odečítáním hodnot z přístrojů a jejich následným exportem do tabulkového procesoru, ve kterém se provede výpočet. Tento přístup by měl také zamezit náhodným chybám při odečítání hodnot a jejich následném přepisování do počítače. Zároveň bych rád vytvořil řešení, které bude přenositelné a aplikovatelné i v jiných laboratořích zabývajících se touto problematikou.

2 Měření pístových pipet

V této kapitole je popsaná základní teorie gravimetrického způsobu kalibrace. Je to z důvodu lepšího pochopení požadovaného řešení. Dále je zde popis stávajícího postupu kalibrace a základní stanovení cíle této práce.

2.1 Analýza normou stanovené kalibrační metody

Kalibrace pístových pipet se provádí vážením objemu destilované vody vypuštěné z pístových pipet. Tento postup se opakuje desetkrát, jak stanovuje norma [1]. Během vážení je použito tárování, to znamená, že každá hodnota z vážené hmotnosti je při následujícím vážení převedená do táry (vynulování hodnoty na váze). K výpočtu objemu je potřeba znát hodnotu hustoty destilované vody a její teplotu při vážení. Zjištěná hmotnost se koriguje na vztlak vzduchu, jelikož i zde platí Archimédův zákon, který u takto precizních měření nelze zanedbat. Pro korekci tohoto vztlaku potřebujeme znát hustotu vzduchu, která se vypočítá z atmosférického tlaku, teploty a vlhkosti vzduchu.

Pro kalibraci pístových pipet gravimetrickou metodou jsou potřebné měřidla s těmito parametry:

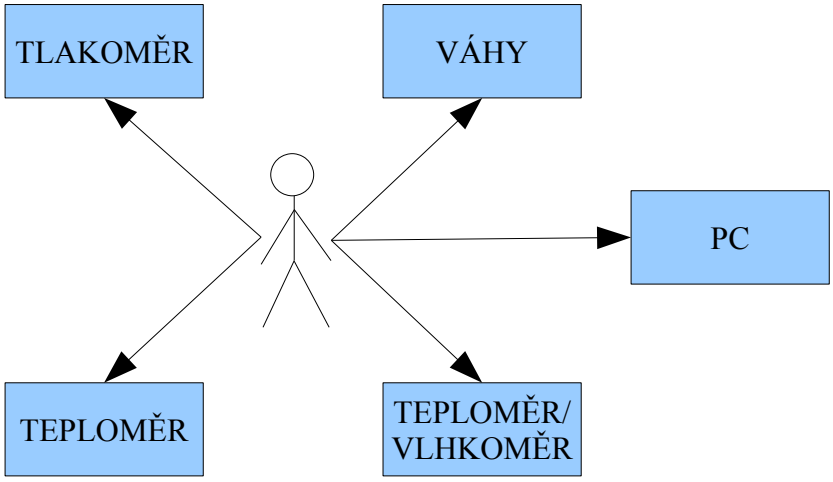
- etalonové váhy 1. třídy přesnosti, s nejmenším dílkem 10^{-2} mg
- digitální teploměr, rozsah 0 °C až 50 °C s dělením 0,01 °C
- barometr, s dělením 0,1 kPa
- digitální teploměr s přesností 0,1 °C a vlhkoměr, s rozlišením 1 % RH

Objem V_i, který odpovídá zjištěné hmotnosti m_i, se vypočítá pomocí korekčního faktoru Z, který v sobě zahrnuje odpovídající hustotu destilované vody pro změřenou teplotu během vážení a korekci na vztlak vzduchu.

2.2 Stávající postup a stanovení cíle této práce

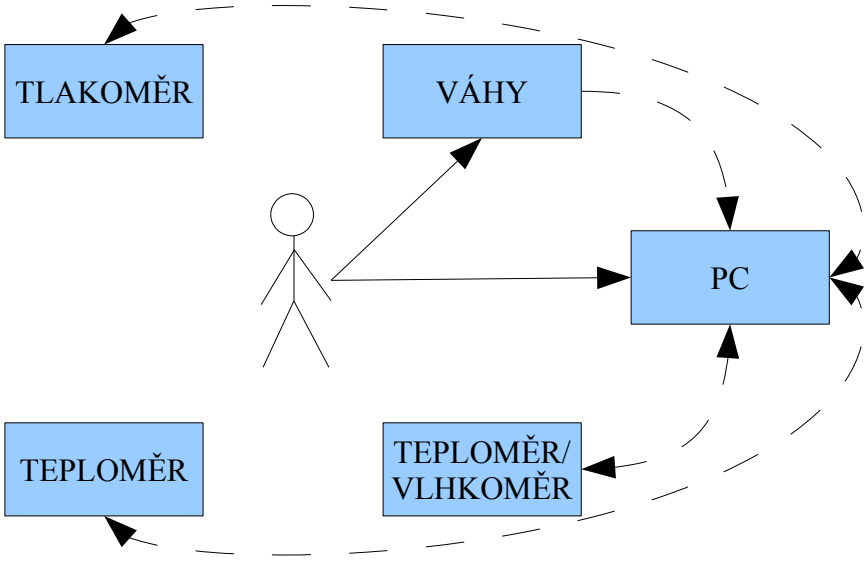
Nynější postup zobrazuje diagram na obrázku 2.1. Laborant, který je na diagramu zobrazen uprostřed, při kalibraci pístových pipet, provádí měření na váze. Podle normalizovaného postupu je prováděno deset opakovaných měření. Při každé iteraci měření se zaznamenávají jednotlivé hodnoty ze všech přístrojů do zápisníkového bloku. To znamená čtyřicet odečtení hodnot, zápis do bloku a jejich následný přepis do připraveného sešitu v aplikaci Excel společnosti Microsoft. Tento obtížný způsob se v praxi zjednodušuje a hodnoty tlaku, vlhkosti a teploty okolí se odečítají dvakrát – na

začátku měření a na konci měření, jelikož tyto hodnoty se příliš nemění vzhledem k tomu, že se měření provádí v klimatizované laboratoři. Poté se vypočítá z těchto hodnot aritmetický průměr a ten je zahrnut do výpočtů. I přes toto zjednodušení se odečítá pro jednu kalibraci dvacet šest hodnot. Tento postup je značně neefektivní a měření může zatížit chybou při přepisování hodnot. Z časového hlediska to znamená, že přibližně deset minut probíhá praktická část, tedy vážení a přibližně dalších deset minut trvá přepis a další operace v počítači.



Obrázek 2.1: Znázornění současného postupu

Cílem této práce je zjednodušit tento proces vytvořením specializovaného programového vybavení, které bude tento proces snímání hodnot a exportování hodnot automatizovat. Výsledný postup práce zobrazuje diagram na obrázku 2.2.



Obrázek 2.2: Výsledný postup

Laborant spustí aplikaci, která je cílem této práce a nastaví parametry pipety, popřípadě změní nastavení aplikace (např. počet měření nebo adresář výstupu). Následně spustí měření. Program od

této chvíle očekává data od vah. Laborant začne provádět měření a každou hodnotu odesílá do počítače pomocí speciálního tlačítka tisku, které se nachází na vahách. Program po obdržení dat odečte hodnoty z ostatních přístrojů. Po dokončení všech měření program zobrazí naměřené hodnoty a uživatel má možnost vybrat hodnoty, které vyhodnotí jako nevyhovující a je třeba je znovu změřit. Pokud jsou naměřené hodnoty v pořádku, uživatel zvolí export do sešitu aplikace MS Excel, ve které se provedou veškeré výpočty. Pro tento účel bude existovat vzorový sešit s předem připravenými výpočty, který se pro každou kalibraci zkopíruje, a budou do něj uložená načtená data. Toto řešení by mělo odstranit možnost vzniku náhodné chyby při opisu hodnot a v neposlední řadě zrychlit celý proces kalibrace pístových pipet. Detailní popis požadavků aplikace je v kapitole 7.

3 Dostupná komerční zařízení

Z důvodu návrhu přenositelné aplikace jsem analyzoval vlastnosti dostupných zařízení, u kterých je pravděpodobné zařazení do měřícího systému. Zejména jaké využívají datové rozhraní pro připojení k počítači a způsob komunikace s těmito periferiemi. Tato kapitola se proto zabývá analýzou dostupných zařízení na trhu, které splňují požadavky normy a zároveň jsou vhodné pro užití v kalibračním softwaru, který snímá aktuální hodnoty. Průzkum trhu byl proveden pro náš trh a pro zjištění názvů výrobců, byly použity katalogy největších dodavatelů laboratorní techniky, jako jsou MERCK spol. s r.o., VITRUM Praha spol. s r.o., Fisher Scientific, spol. s r.o.. Po nalezení vyhovujících přístrojů v katalogu těchto dodavatelů, byla prozkoumána kompletní nabídka daného výrobce, včetně specifikací datové komunikace jednotlivých vhodných přístrojů.

3.1 Váhy splňující požadavky metodiky

Protože se jedná o nejstěžejnější a nejdražší zařízení pro kalibraci pipet, provedl jsem porovnání všech dostupných výrobců. V současnosti je na evropském trhu možné pořídit analytické váhy v této kategorii přesnosti od výrobců *Kern & Sohn, METTLER TOLEDO, Ohaus, RADWAG, Sartorius*. Všichni vyjmenovaní výrobci implementují do svých zařízení datové rozhraní, které je většinou realizované pomocí RS-232. U vyšších modelů bývají dostupná i rozhraní Ethernet a USB. Popis datové komunikace je u všech zmíněných výrobců volně dostupný a lze ho tak využít pro implementaci vlastních aplikací. Každý výrobce má ale vlastní komunikační protokol. Kromě těchto protokolů, výrobci *METTLER TELEDO* a *Sartorius,* využívají u nových nejvyšších řad analytických vah komunikační protokoly zvané *MT-SICS* (*METTLER TOLEDO - Standard Interface Command Set*) a *Sartorius SICS Interface*, které jsou přehledově popsány v podkapitolách 3.1.1 a 3.1.2. U většiny vah výrobce také implementuje standardizovaný tisk výsledků splňující *GLP (Good Laboratory Practice)*. Ten by mohl být pro účel získávání dat, také použit. Umožnil by ale jen jednosměrnou komunikaci, hodnoty lze získat pouze vyvoláním tisku na vahách.

3.1.1 Protokoly METTLER TOLEDO MT - SICS a Sartorius – SICS

Společnosti *METTLER TOLEDO* a *Sartorius* implementují do svých nových vah množinu příkazů SICS, díky kterým lze ovládat váhy přes datové rozhraní. Ačkoliv zkratka SICS je u obou společností totožná, nejedná se o dva totožné protokoly. I názvy se liší. V případě společnosti *METTLER TOLEDO* se jedná o zkratku „*Standard Interface Command Set“*, kdežto u spol. *Sartorius*

zkratka znamená „*Standard Interface Common Set*“. Pravděpodobně z tohoto důvodu nelze v době psaní této práce nalézt obecný standard popisující tuto množinu příkazů. Nicméně v obou případech má určitá podmnožina příkazů stejnou syntaxi i sémantiku. Oba výrobci příkazy dělí do několika úrovní. *Sartorius* dělí příkazy do tří úrovní [14] a *METTLER TOLEDO* do čtyř úrovní [15]. Úrovně *Level 0* a *Level 1* jsou u obou výrobců stejné a to jak množinou příkazů, tak i verzí 2.3x pro *Level 0* a verzí 2.2x pro *Level 1*. Tyto dvě úrovně obsahují příkazy pro základní manipulaci s váhami. Například odeslání aktuální vážené hodnoty v případě ustálení vah, nebo hodnoty bez ustálených vah, sériové číslo vah, vytisknutí zprávy na displej vah apod. (přesný popis všech příkazů je v návodech [14] a [15]). Tyto dvě úrovně by měly být implementovány všemi novými řadami vah, od nejjednodušších vah až po pokročilé vážící stanice. Zbývající jedna resp. dvě úrovně jsou již vázány na vážící programy, které jsou součástí pokročilejších vah, a jejich popis závisí na konkrétním modelu.

3.1.2 ISO/GLP tiskový výstup / záznam

Všechny váhy, které vyhovují výše uvedeným požadavkům, umožňují tisk výsledků pomocí tiskáren s rozhraním RS-232, které se prodávají jako volitelné příslušenství. Tiskový výstup je koncipován tak, aby splňoval doporučení *GLP* (*Good Laboratory Practice*) – obsahuje GLP hlavičku a GLP patičku. V GLP hlavičce je uvedeno datum, čas začátku měření, výrobce vah, model, sériové číslo vah, verze firmwaru vah a identifikační číslo měření. GLP patička obsahuje čas a datum ukončení měření a pole pro podpis osoby provádějící měření. V případě, že by u některých vah nebyl dostupný popis komunikačního protokolu a váhy by umožňovaly tiskový výstup, bylo by možné jej využít k získání hodnot. Avšak tento způsob by byl jen jednosměrný – programově by nešlo hodnoty vyčítat, ale pouze jen čekat na zaslání hodnoty z vah.

3.2 Teploměry splňující požadavky metodiky

I přes poměrně vysoký počet výrobců, kteří jsou dostupní na našem trhu, požadavky na přesnost a možnost připojení k počítači pro snímání aktuálních hodnot, seznam značně zredukovaly na produkty společností *Greisinger electronic GmbH* (dále jen *Greisinger*) a české společnosti *Comet System, s.r.o.* (dále jen *Comet*). Oba dva výrobci poskytují ruční digitální teploměry s připojitelnými teplotními sondami se senzorem *Pt100 / Pt1000,* nebo sondy s termočlánky *K*. Společnost *Greisinger* nabízí ke svým produktům placenou proprietární knihovnu pro vývoj vlastních aplikací a samotný způsob komunikace nepopisuje. Pro připojení přístrojů nabízí galvanicky oddělenou kabeláž s převodníkem na USB, anebo s výstupem na rozhraní RS-232. U společnosti *Comet* lze nalézt produkty, které mají výstupní rozhraní RS-232, RS-485 a Ethernet. Jako komunikační protokoly

implementují pro sériová rozhraní *Modbus RTU, ADAM firmy Advantech, nebo ARION firmy AMIT* [16] a pro Ethernetové rozhraní *Modbus TCP, SNMP, SOAP* (*Simple Object Access Protocol*) a také přes webové rozhraní snímače [16].

3.3 Barometry splňující požadavky metodiky

Většina dostupných zařízení se prodávají jako kombinovaná zařízení. Integrují se s teploměry a případně i s vlhkoměry. Takovéto zařízení na náš trh distribuují společnosti *Greisinger, Comet*, *AHLBORN měřicí a regulační technika spol. s r.o.* (dále jen *Ahlborn*). Společnosti *Greisinger* a *Comet* poskytují stejné možnosti připojení a komunikace, které byly popsány v předchozí podkapitole.

Společnost *Alhorn* umožňuje u vybraných typů připojení pomocí RS-232, USB a Ethernetem. Bohužel při prohledávání se mi nepodařilo zjistit, jaký komunikační protokol podporuje, zda má vlastní a volně dostupný protokol, nebo zda má uzavřený proprietární protokol a distribuuje pouze knihovnu ať již placenou, či neplacenou. Při prohledání našeho trhu bylo zjištěno, že barometry v podobě samostatného přístroje je u nás možné pouze pořídit od společnosti *GE Measurement & Control Solutions* a to pouze jediný typ - přesný barometr *DPI 142*. Tento typ implementuje implicitně RS-232 a volitelně *IEEE 488.2 (GPIB)*. Jako komunikační protokol využívá *SCPI* (Standard Commands for Programmable Instruments), který byl definován v roce 1990 s IEEE 488.2. Tento standard definuje příkazy a jejich syntaxi, které slouží pro ovládání měřidel. Standard není vázaný na konkrétní typ rozhraní a je proto využitelný nejen pro GPIB, pro který byl vytvořen, ale je aplikovatelný nad RS-232C, USB, Ethernetem a nad jinými rozhraními [17].

3.4 Vlhkoměry splňující požadavky metodiky

Samostatný digitální vlhkoměr s možností softwarového odečítání hodnot se mi v době psaní této práce nepodařilo vyhledat. U všech výsledků vyhledávání se jednalo o kombinované zařízení s teploměrem, a to z důvodu možnosti stanovení rosného bodu. Také je možné najít kombinaci například s barometrem. V této oblasti přesnosti jsou opět k dispozici především výrobky společností Greisinger a Comet. Jejich rozhraní a možnosti komunikace jsou stejné, tak jak byly popsány v předchozí kapitole.

3.5 Shrnutí výsledků průzkumu dostupných zařízení

Z přehledu je patrné, že všechna zkoumaná zařízení implementují sériové rozhraní RS-232C. U některých modelů je možné připojení pomocí USB, které je však většinou řešeno pomocí volitelného příslušenství ve formě převodníku RS-232C na USB. Méně často se vyskytuje možnost připojení pomocí Ethernetu a nejméně pomocí rozhraní IEEE 488 – GPIB. Co se týče popisu komunikačního protokolu, tak se ve většině případů jedná o dobře dokumentované rozhraní, ať již ve formě standardu, tak ve formě komerčního protokolu s dostupným popisem. Výjimku tvoří společnost Greisinger, která poskytuje pouze placenou knihovnu, díky které je možné uskutečnit komunikaci s jejich produkty. Dále za povšimnutí stojí, že kromě vah a přesných digitálních teploměrů, je větší část měřidel v kombinaci s jinými. Tedy je snazší pořídit vlhkoměr v kombinaci s teploměrem, nežli samotný vlhkoměr. Při návrhu aplikace bude nutné s tímto faktem počítat a nespoléhat na fakt, že každé zařízení poskytuje pouze jednu měřenou veličinu.

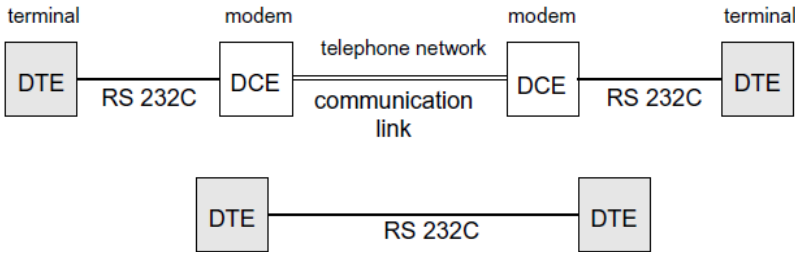
4 Fyzická rozhraní přístrojů

Z výše uvedeného srovnání dostupných zařízení je patrné, že pro datovou komunikaci u zařízení se nejčastěji využívá RS-232c a USB. Proto je zde popis těchto dvou rozhraní.

4.1 Sériová rozhraní

Tato kapitola se zabývá popisem nejrozšířenějších sériových rozhraní, která lze nalézt u zkoumaných přístrojů. Nejdříve je zde popsané rozhraní RS-232C, které většina zkoumaných zařízení využívá. V druhé části kapitoly bude popis rozhraní USB, které lze nalézt především u nových zařízení a také často ve formě převodníku RS-232C na USB, který se prodává jako volitelné příslušenství k přístrojům. USB je bohužel na tolik komplexní, že jeho popis musí být velmi zestručněn a proto neobsahuje všechny informace. Předpokládám, že ale vystihuje ty nejpodstatnější části popisu tohoto rozhraní.

4.1.1 RS-232 / EIA-232 / ITU-TSS V.24, V.28



Obrázek 4.1: Příklad možných variant propojení DTE, DCE[9]

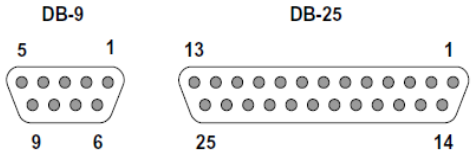
První norma, popisující rozhraní RS-232, pochází z roku 1962 a byla vydána EIA(*Electronics Industries Association*) a popisuje z pohledu vrstvového systému ISO/OSI jen fyzickou vrstvu. Zkratka RS znamená *Recommended Standard* [9]. Další revize byla v roce 1969 a označuje se RS-232C a je také nejrozšířenější (existují i varianty D a E). RS-232 byl vyvinut pro účely propojení dvou koncových zařízení a to mainfraimových počítačů a terminálů. Tato koncová zařízení jsou značená DTE (Data Terminal Equipment). Tyto dva typy zařízení se propojovaly přes telefonní síť a z toho důvodu musely přijít mezi tato dvě zařízení modemy, které jsou v tomto standardu značeny jako DCE (Data Communication Equipment). Pro lepší představu je zde obrázek č. 4.1. Z tohoto účelu jsou pojmenované některé signály resp. piny např. RI (Ring Indicator) i přes to, že v současnosti nemají téměř nic společného s dřívějším užitím RS-232 [10]. Obrázek 4.1 také zobrazuje

propojení dvou DTE zařízení – pro toto propojení je třeba využít tzv. *null modem*. Jedná se o speciální zapojení kabelu. V současnosti běžné PC, které obsahuje sériový port, je typu DTE a tomu i odpovídá typ konektoru, který bude popsán níže.

Jelikož RS-232C je standard pocházející z USA, tak také ITU jej popsala v mezinárodních standardech ITU–TSS V.24 a V.28 [9]. Tyto dva standardy jsou kompatibilní a tedy pokud zařízení splňuje popis jedné normy, tak je téměř vždy kompatibilní i s druhou normou. Dále v textu se bude používat jen termín RS-232, pro jeho lepší zažitost.

RS-232 popisuje tedy signály – jejich napěťové úrovně, časování apod., dále definuje mechanické charakteristiky jako je konektor, popis pinů atd. [7]. Ačkoliv je toto rozhraní značně staré a rychlostí je oproti modernějším sériovým rozhraním, jako jsou USB nebo FireWire, značně pomalé, pro svoji jednoduchost se stále využívá v mnoha přístrojích.

Pro připojení se využívají konektory DB-9 a DB-25 a jsou zobrazeny na obrázku č. 4.2, přičemž RS-232C definuje pouze pětadvacetipinovou variantu konektoru, ale i přes to se převážně se využívá DB-9 s devíti piny. Popis signálů a jejich umístění v konektoru zobrazuje obrázek č. 4.3. Zařízení typu DTE mají obvykle samce a zařízení DCE naopak samice konektoru, tedy konektory, které mohou být v PC, jsou typu samec.



Obrázek 4.2: Konektory využívané u RS-232[9]

Co se týče úrovní signálů, tak RS-232C specifikovaný rozsah hodnot je velký. Vstupy musí být schopny přijmout logickou jedničku, která je reprezentovaná napětím od -3 V do -15 V a logickou nulu, která je reprezentovaná napětím od 3 V do 15 V.

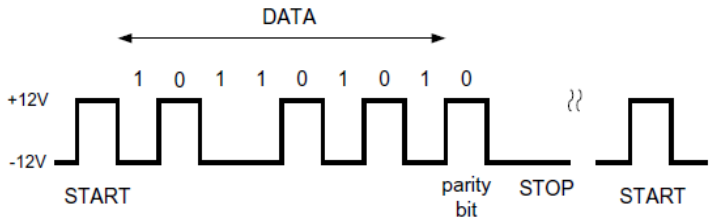
Pin No DB-9	Pin No DB-25	Code	Description
-	1	PG	Protective Ground
5	7	SG	Signal Ground
3	2	TxD	Transmitted Data
2	3	RxD	Received Data
7	4	RTS	Request to Send
8	5	CTS	Clear to Send
6	6	DSR	Data Set Ready
4	20	DTR	Data Terminal Ready
1	8	DCD	Data Carrier Detected
9	22	RI	Ring Indicator
	15	DB	Transmitter signal timing
	17	DD	Receiver signal timing
	24	DA	Transmitter signal timing

Obrázek 4.3: Seznam signálů v konektorech RS-232[9]

Dále norma udává, že kapacita na výstupu nesmí přesáhnout 2500 pF a rezistence musí být v rozsah 3000 Ω až 7000 Ω [12]. RS-232 může fungovat, jak synchronně, tak asynchronně, přičemž asynchronní varianta je nejběžnější. V synchronní variantě musí být vyveden hodinový signál jednoho ze dvou zařízení a všechny přenášené bity jsou pomocí tohoto signálu synchronizovány. Je tedy přesně stanoveno, kdy se mají data vystavit a kdy číst. Signál může být veden samostatným vodičem nebo může být odvozen z dat. Specifikace, zdali je systém citlivý na hranu či na hladinu záleží na konkrétním výrobci. Tento hodinový signál může mít stálou frekvenci nebo se může měnit v nepravidelných intervalech [11].

I přesto, že synchronní přenos je více efektivní (např. větší datové rámce, rychlost), běžnější je asynchronní přenos. Pro zajištění správného přenosu dat, je třeba zajistit synchronizaci vnitřních hodin. K tomu slouží start a stop bity. Ty s daty vytváří tzv. *SDU – Serial Data Unit*, kde první bit je start bit, následuje sedm nebo osm datových bitů, za nimi může být paritní bit a na konci je jeden nebo dva stop bity. SDU zobrazuje obrázek č. 4.4.

Paritní bit může reprezentovat sudou nebo lichou paritu, popřípadě může být space (vždy log 0) nebo mark (vždy log 1).



Obrázek 4.4: příklad SDU[9]

Je nutné, aby obě strany měly nastavené stejné parametry přenosu, tedy rychlost přenosu a formát SDU. Rychlost přenosu se udává v baudech. Modulační jednotka baud (zkr. Bd) udává počet změn za sekundu a konkrétní rychlost přenosu dat je nutné vypočítat s využitím znalosti velikosti SDU. RS-232C nespecifikuje rychlosti přenosu. Obvykle jsou brány řadou 150, 300, 600, 1200 až do 115200 baudu. Při použití převodníků USB na UART, rychlosti mohou dosahovat až 921,6 kBd. Maximální délka kabeláže je neúměrná rychlosti přenášených dat. Například se udává, že při rychlosti 19200 baudů je maximální délka 50 m, pro 4800 baudů 100 m, ovšem standard RS-232c garantuje rychlost 19800 baudů jen pro délku 15 m. Existuje také auto-detekční systém rychlosti.

Výměna dat může probíhat ve třech režimech. Prvním z nich je simplex mód. V tomto režimu jsou data vysílána jen v jednom směru. Další možností je tzv. half-duplex. Při tomto režimu jsou data vysílána oběma směry a využívá se časového multiplexu pro střídání směru toku dat. Poslední variantou je full-duplex, při kterém se vysílá oběma směry zároveň (z toho plyne nutnost dvou datových vodičů) [9].

Aby nedocházelo k přetížení zařízení při rychlejší výměně dat, než je dané zařízení schopno zpracovat, je RS-232 vybavený handshake, tedy prvkem, který může řídit datové toky (zjišťováni připravenosti odesílat/přijímat data). Může být hardwarový nebo softwarový. Hardwarový je zajištěný propojením příslušných signálů v konektorech mezi zařízeními (obvykle RTS a CTS signály viz obrázek č. 4.3). Softwarový je tvořený speciálními bajty XON a XOFF. XOFF odesílá příjemce a sděluje tím odesílateli, aby zastavil odesílání dat, dokud neobdrží XON [10].

4.1.2 USB - Universal Serial Bus

Tato sběrnice byla vyvinuta za účelem zjednodušení připojování periferií. První verze standardu USB 1.0 byla vydána v roce 1996 [21] a z dnešního pohledu, ve většině případů USB úspěšně nahradilo běžná starší sériová (RS-232C, IBM PS/2) a paralelní rozhraní (IEEE 1284/Centronics). Vzhledem k tomuto faktu, sériové rozhraní RS-232C úplně vymizelo z notebooků a mnoho výrobců základních desek tyto porty přímo nevyvádí a pokud jej vyvede, vyvede pouze jeden port. Je tedy velmi pravděpodobné, že většina měřidel bude připojena přes USB. Jelikož toto rozhraní mají jen novější měřící zařízení a jak bylo popsáno v kapitole 3, většina zařízení užívá RS-232C, budou se tyto zařízení pravděpodobně připojovat pomocí převodníku RS-232C na USB, který se v operačním systému jeví jako virtuální sériový port. Oproti většině starších rozhraní USB přináší možnost zapojovat a vypojovat periferie za chodu, počet připojených zařízení může být až 127 na jeden USB host (řadič), poskytuje větší přenosovou rychlost a standard specifikuje i maximální proud, který může port dodat pro napájení periferii. V době psaní této práce je nejnovějším standardem USB 3.0 a rozdíly mezi jednotlivými verzemi jsou v tabulce č. 1.

Verze USB	Rok prvního vydání	Teoretická maximální rychlost [Mb/s]	Označení rychlosti	Maximální proud jednoho portu [mA] High/Low Power
USB 1.x	1996	1,5	*Low Speed*	500 / 100
		12	*Full Speed*	
USB 2.x	2000	480	*High Speed*	500 / 100
USB 3.0	2008	4800	*Super Speed*	900 / 150

Tabulka 1: Tabulka verzí USB a jejich parametrů, data byla čerpána z [21]

Fyzická topologie je víceúrovňová hvězdicová, ve středu každé hvězdy se nachází USB rozbočovač (USB hub), který má typicky dva, čtyři, nebo sedm portů. Standard stanovuje maximální úroveň na pět, to znamená, že lze v sérii zapojit pět USB rozbočovačů a na jeden USB host lze maximálně připojit 127 zařízení, mezi které se počítá i USB hub. Na logické úrovni se ale pracuje s lineárním uspořádáním – z programátorského hlediska je fyzická topologie skryta a hostitelské

aplikace nepotřebují znát počet rozbočovačů v cestě k zařízení [21]. Specifikace USB také určuje kabeláž a konektory tak, aby se zabránilo cyklickému zapojení [18].

Hostitelské zařízení, typicky počítač, obsahuje USB Host (řadič), na který je připojen kořenový rozbočovač (*Root hub*), který se také započítává do výše zmíněného počtu sériově zapojených rozbočovačů. Je zde také zajištěna zpětná kompatibilita - novější standard podporuje starší rychlosti. Je tak možné na rozbočovač verze USB 3.0 zapojit rozbočovač USB 2.0 a na něj zařízení s USB 3.0, ale maximální rychlost komunikace bude omezena na *High Speed*, tedy maximální rychlost USB 2.0. Rychlosti uvedené v tabulce č. 1 obsahují maximální teoretické rychlosti, které jsou v praxi menší díky provozní režii. V přehledu rychlostí vybočuje rychlost označená *Low Speed*, která má maximální teoretickou rychlost 1,5Mb/s a je určena pro pomalá zařízení, jako jsou zařízení HID (klávesnice, myš) a také většina měřidel. Všechna zařízení využívají stejnou sběrnici a reálná rychlost závisí na mnoha faktorech, jako jsou například: vytíženost sběrnice, chyby, synchronizace aj. Z tohoto důvodu nemusí být rychlost vždy predikovatelná. I navzdory tomuto faktu, USB podporuje i aplikace, které mají přenosy dat citlivé na čas a garantuje rychlost nebo maximální zpoždění dat (viz níže isochronní přenosy a přenosy s přerušením). Z výkonnostních důvodů obvykle počítač obsahuje více USB řadičů [21].

Jako přenosové médium se využívá standardem daná metalická kabeláž nebo také existuje bezdrátová varianta *Wireless USB* (pracující v bez licenčním pásmu 2,4 GHz), která je ale oproti kabeláži méně rozšířená a proto se jí nebudu nadále zabývat. Kabel pro USB verze 1.x a 2.x se sestává ze čtyř vodičů: V_{BUS}, GND, D+ a D-. Vodiče D+ a D- jsou diferenční datové vodiče, vedené formou kroucené dvojlinky - technika *TMDS (Transition Minimized Differential Signaling)*. V těchto paralelních vodičích jsou logické stavy vůči sobě invertované a na přijímací straně se bere hodnota rozdílu těchto dvou signálů. To umožňuje vyšší odolnost vůči elektromagnetickému šumu. Tento přenos je polo duplexní. Jako kódování se užívá *NRZI (Non Return To Zero Inverted)*. V literatuře se uvádí označení napěťové úrovně jako K, pro nízké napětí a J pro vysokou hodnotu [21].

Metoda mapování binárních hodnot na napěťové úrovně NRZI reprezentuje logickou 0 jako změnu J na K a naopak. Logická 1 se reprezentuje tak, že úrovně napětí na signálových vodičích jsou beze změny. Zároveň každou změnou dochází k takzvanému *Clock recovery* – synchronizace hodin periferie [22]. Z tohoto důvodu se za každých šest bitů vkládá jeden nulový, aby došlo ke změně úrovně a synchronizaci hodin i v případě, že by se vyskytlo šest jedniček za sebou - tzv. *Bit-stuffing*. Sedm jedniček za sebou, se považuje za chybu [21].

Vodič označený V_{BUS} a GND slouží pro napájení periferií. Mezi těmito vodiči je stejnosměrné napětí v toleranci 4,45 V až 5.25 V. Maximální proud, který může USB Host dodat, se liší pro jednotlivé verze USB standardu a jejich hodnoty jsou uvedeny v tabulce č. 1. Jsou zde uvedeny dvě hodnoty pro každou verzi USB – High a Low Power. Hodnoty Low Power jsou pro USB hosty v

zařízeních, které fungují v energeticky úsporném režimu, například jsou napájeny z akumulátoru. Hodnoty High Power jsou pro USB Host v běžném zařízení, jako je stolní počítač, který má možnost dodat potřebný proud.

Nový standard USB 3.0 přinesl i změny do konektorů a počtu vodičů rozhraní. I přesto je zachována zpětná kompatibilita se staršími verzemi. USB 3.0 využívá nové architektury se dvěma fyzickými sběrnicemi, které pracují paralelně a navíc z důvodu zpětné kompatibility má USB 3.0 i dvojici vodičů D+ a D- ze sběrnice USB 2.0. Je tedy celkem deset vodičů: V_{BUS}, D+, D-, GND, SSRX−, SSRX+, SSTX−, SSTX+, GND_DRAIN (stínění) a USB OTG ID. Jak již bylo zmíněno výše, novinkou jsou dvě nezávislé sběrnice (označené SSRX± a SSTX±) a je tak možné vysílat a přijímat data zároveň. Další novinkou je, že koncové body mohou informovat asynchronně USB host o připravenosti jejich dat (dříve se USB host periodicky dotazoval na dostupnost dat) [21].

Mezi USB zařízení se řadí rozbočovače a periferie. Jednotlivé typy zařízení jsou rozděleny do kategorií a podkategorií. Například kategorie *Communications* má podkategorie *Direct Line, Ethernet Networking* aj. a například převodníky RS232 na USB spadají do kategorie *CDC Data*. Stručný seznam kategorií a podkategorií je například v [18].

Každé zařízení obsahuje deskriptor, ve kterém je uvedeno číslo výrobce (Vendor ID), číslo produktu (Product ID), třída, podtřída, požadavky na maximální proud, sériové číslo a mnoho dalších údajů [21].

Pro řízení toku dat na sběrnici je klíčový řadič – *USB host*, který veškerou komunikaci řídí a rozhoduje o přiřazení času sběrnice jednotlivým zařízením. USB neumožňuje *broadcast* a řadič se periodicky dotazuje jednotlivých zařízení, zdali mají připravená data k přenosu. Do verze USB 3.0, zařízení nemohlo řadiči vyslat žádná data, aniž by bylo dotázáno řadičem, tedy periferie nemůže ani poslat zprávu o připravenosti dat k odeslání. Od verze 3.0 může zařízení dát řadiči vědět, že má připravená data k odeslání [21]. Každému zařízení USB řadič dynamicky přiřadí unikátní adresu, pomocí které s touto periferií komunikuje.

Jsou zde čtyři možné typy přenosu – *řídící (control), objemný (bulk), přerušením (interrupt)* a *isochronní (isochronous)* [21].

Všechna zařízení musí podporovat řídící přenos, který je určený pro zjištění aktuálního stavu zařízení, nastavení zařízení, přiřazení adresy a jiné řídící činnosti [21].

Přenos typu „*bulk*“ je určený pro přenos objemných dat, pro které není klíčovou vlastností rychlost, která může klesnout v případě, že je sběrnice velmi vytížená. V případě, že sběrnice není zaneprázdněná, jde o nejrychlejší přenosy. Tento typ přenosu není podporován u zařízení s komunikační rychlostí stanovenou na Low Speed. Jedná se například o diskové operace, data ze scanneru a přenos dat do tiskárny [21].

Přenosy s *přerušením* jsou určeny pro zařízení vyžadující nízkou latenci, či zpoždění, anebo potřebují být periodicky kontrolována. To znamená, že tento typ přenosu má garantované maximální zpoždění. Toto zpoždění je v rozsahu 10–255 ms pro Low Speed, pro 1–255 ms Full Speed a 125 μs až 4,096 s pro High Speed a Super Speed. Zařízení s libovolnou výše uvedenou rychlostí mohou tento typ přenosu využít. Pro zařízení, které pracují v rychlostním režimu Low Speed, je tento způsob přenosu jediný, který lze použít. Příkladem zařízení, která využívají tento typ přenosu, jsou HID zařízení [21].

Posledním typem přenosu je *isochronní*, který je vhodný pro zařízení, u kterých se využívá datový proud, jako jsou např. audio-video zařízení. Isochronní přenosy mají garantovanou přenosovou rychlost za předpokladu, že USB host může v pravidelných intervalech požadovat od zařízení specifikovaný počet bajtů. Nejmenší interval je stanovený na 1 ms pro Full Speed a 125 μs pro Super Speed. U těchto přenosů není korekce chyb. Ucelený přehled a porovnání jednotlivých typů přenosů je v příloze č. 4.

5 Popis přístrojů užívaných na ČMI Brno

Tato část se zabývá technickými specifikacemi přístrojů, které jsou využívány při kalibraci pístových pipet na ČMI Brno. Je zde popsaný aplikační protokol i základní parametry přístrojů, včetně továrních nastavení, které budu užívat jako výchozí hodnoty pro implementaci.

5.1 Elektromechanická váha Sartorius AG - CPA225D

Klíčovým prvkem celé měřící soustavy jsou váhy spadající do první třídy přesnosti. Jak již bylo uvedeno dříve, signál vyslaný z těchto vah je spouštěcím signálem pro odečtení hodnot z ostatních zařízení.

5.1.1 Popis zařízení

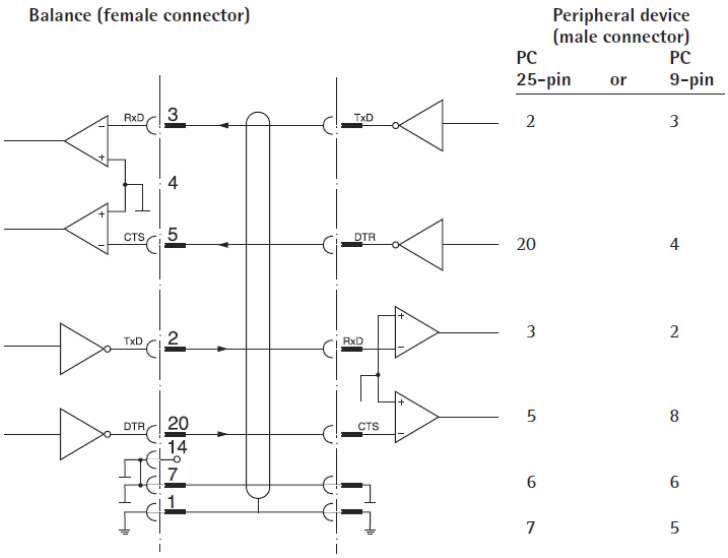
Přesný typ a jméno výrobce je CPA225D-OCE Sartorius AG, výrobní číslo 18901986. Váhy mají horní mez vážitelnosti 80/220 g s rozlišitelností 0,01 mg / 0,1 mg. Datové rozhraní je RS232C-S / V24-V28.

5.1.2 Popis rozhraní

Tovární parametry rozhraní jsou nastaveny na 1200 baudů, lichou paritu, 1 stop bit a handshake je stanovený na dva znaky po signálu CTS. Kompletní parametry jsou uvedeny v příloze č. 5 a také v [3].

Parametry se dají nastavovat přímo na váhách, nebo softwarově. Pro hardwarový handshake je potřeba propojit piny signálů CTS-DTR oboustranně.

Rozhraní na straně vah je vyvedeno do samice dvacetipětipinového konektoru D-submini (DB25S). Originální kabeláž od výrobce nebyla součástí dodávky. Obrázek 5.1 zobrazuje doporučené zapojení kabeláže, podle které bude kabel v rámci implementace zhotoven. Výrobcem maximální doporučená délka kabeláže je 15 m.



Obrázek 5.1: Zapojení konektorů rozhraní vah [3]

5.1.3 Popis komunikačního protokolu

Výstup dat je možné vyvolat dvěma způsoby. První je pomocí speciálního tlačítka tisku na vahách a druhá možnost vyvolání výstupu je pomocí zaslání zprávy z připojené periferie – v tomto případě počítače. Sekvence znaků na výstupu koresponduje se znaky, které jsou zobrazeny na displeji vah (počet mezer místo číslic). Dále se dají výstupy rozdělit do tří kategorií. Prví kategorie specifikuje data při běžném provozu, druhá speciální stavy a třetí chybové stavy. Všechny znaky jsou kódovány v 7 bit ASCII. Maximální počet hodnot, které jsou schopny váhy odeslat je deset za vteřinu.

Popis šestnácti znakového výstupu: Tabulka č. 2 zobrazuje formát přijímaných dat pro všechny tři situace, které byly popsány výše. Každý řádek tabulky zobrazuje právě jeden z těchto typů výstupu. Ve sloupcích jsou prezentovány jednotlivé znaky výstupu.

Běžný typ výstupu je reprezentovaný prvním řádkem a praktická ukázka je v tabulce č. 3. První sloupec tohoto řádku reprezentuje znaménko. To může nabývat hodnot +, - a také může být nahrazeno mezerou. Od pátého do desátého znaku je uložena samotná hodnota v podobě číslic a jedním z těchto znaků je vždy desetinná tečka, která je plovoucí. Počet číslic je proměnlivý v závislosti na pozici desetinné tečky. Nadbytečné nulové číslice před desetinnou tečkou jsou nahrazeny znakem mezery. Po číselné hodnotě následuje mezera a na jedenácté až třinácté pozici je reprezentovaná jednotka. Výchozí jednotkou továrního nastavení je gram, který je reprezentovaný znakem 'g' a je uložen na dvanácté pozici. Obsáhlý seznam možných, převážně nemetrických jednotek, je uveden v [3]. Následují znaky CRLF značící konec dat.

Řádek S reprezentuje další kategorii zpráv a to speciální zprávy. Tento typ zprávy je reprezentovaný pouze dvěma znaky. První je na sedmé pozici a může nabývat hodnot 'H' při přetížení váhy, 'L' při příliš malé zátěži a 'C' při kalibraci nebo nastavování. Následuje znak mezery.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
B	Z				Č	Č	Č	Č	Č	Č	M	J	J	J	CR	LF
S							S	M								
Ch				'E'	'r'	'r'	M	Č	Č	Č	M	M	M	M	CR	LF

Tabulka 2: Zobrazení jednotlivých pozic znaků pro 16 znakovou zprávu

Z – znaménko, M – mezera, Č – číslice, J – jednotka, S – stav, 'x' – x je znak vždy na stejné pozici, CR – Carriage return, LF – Line feed, ***B*** *– běžná zpráva s hodnotou váhy,* ***S*** *– speciální zpráva,* ***Ch*** *– chybová zpráva*

Na řádku Ch je zobrazený formát chybových zpráv. Čtvrtou až šestou pozici vždy zabírají znaky 'E', 'r', 'r'. Následují znak mezery a tři číslice, které určují typ chyby. Seznam chyb je uvedený v příloze. Dále jsou čtyři mezery a konec dat reprezentovaný CRLF.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
+					1	2	3	.	4		g			CR	LF

Tabulka 3: Příklad odeslaných dat

Popis dvaadvacetiznakového výstupu (výchozí nastavení vah je v tomto formátu):

Tento formát dat rozšiřuje předchozí o šest znaků, které jsou přidány před tento původní formát, jak je zobrazeno v tabulce č. 4. Těchto šest znaků je ID kód, který se mění v závislosti na zvoleném programu vážení. Kompletní seznam jednotlivých ID kódů je uvedený v příloze. Zde je uvedený stručný seznam kódů: *Stat* – stav, *N* – Netto N, *T1* – tára 1.

Hodnoty běžných datových zpráv v továrním nastavení začínají znakem N, následují mezery do šestého znaku. Sedmý znak znamená znaménko a následující znaky jsou totožné s popisem pro šestnáctiznaková data. Příklad je uveden v tabulce č. 5.

U speciálních zpráv se situace liší, oproti šestnáctiznakové zprávě, ve dvou vlastnostech. První z nich je to, že první čtyři znaky jsou obsazeny znaky 'S', 't', 'a', 't'. Druhou odlišností je, že zde není zpráva typu 'C'. Hodnota se nachází na třinácté pozici a následuje mezera.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
B	I	I	I	I	I	I	Z	M	Č	Č	Č	Č	Č	Č	Č	Č	M	J	J	J	CR	LF
S													S	M								
C				'E'	'r'	'r'	M	Č	Č	Č	M	M	M	M	CR	LF						

Tabulka 4: Zobrazení jednotlivých pozic znaků pro 22 znakovou zprávu

Popis tabulky:

I – identifikátor, Z – znaménko, M – mezera, Č – číslice, J – jednotka, S – stav, 'x' – x je znak vždy na stejné pozici, CR – Carriage return, LF – Line feed

B – běžná zpráva s hodnotou váhy, **S** – speciální zpráva, **C** – chybová zpráva

U chybových zpráv je opět na prvních čtyřech znacích řetězec *Stat*. Následují znaky mezer až do desáté pozice a na jedenácté pozici začíná chybová hláška. Následující znaky jsou totožné s šestnácti znakovou variantou.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
P	N						+				1	2	3	.	5	6		g			CR	LF

Tabulka 5: Příklad formátu zaslaných dat

Popis formátů příkazů, které lze zaslat vahám:

jsou stanoveny dva formáty, pomocí kterých lze ovládat váhy, a jsou zobrazeny v tabulce č. 6.

	1	**2**	**3**	**4**	**5**
Formát 1	Esc	!	CR	LF	
Formát 2	Esc	!	N	CR	LF

Tabulka 6: Formát příkazů

Popis tabulky:

Esc – escape, ! – příkaz, N – číselná část příkazu, CR – Carriage return

První formát je složen ze čtyř znaků a jediný variabilní znak je na druhé pozici. Tímto znakem jsou reprezentovány jednotlivé příkazy. Kompletní seznam příkazů je uveden v [3]. Například hodnota 'P' vyvolá stejnou událost jako tlačítko tisku, 'O' zablokuje klávesy na váhách a 'R' je opět odblokuje.

Druhý formát se liší od předchozího, přidáním jednoho znaku za druhou pozici a celková velikost zprávy je pak pět znaků. Tento znak nabývá vždy číselných hodnot [3] a specifikuje další příkazy pro váhy. Například dvojice řídících znaků 'x ' a '0' vyvolá vnitřní kalibraci vah.

5.2 Číslicový tlakoměr Druck DPI 140

Při kalibraci se hodnoty atmosférického tlaku odečítají z Druck DPI 140 a využívají se pro výpočet korekce vztlaku vzduchu.

5.2.1 Popis zařízení

Barometr Druck DPI 140 byl vyroben v Druck Limited, UK a má výrobní číslo 14100988. Měřící rozsah udává výrobce 800 až 1150 hPa se specifikovanou přesností 0,15 hPa.

5.2.2 Popis rozhraní

Tento přístroj má dvě rozhraní: RS232c, IEEE488. Z důvodu, že řadiče IEEE 488 jsou oproti RS232 několikanásobně dražší a navíc v této aplikaci by nebyly využité výhody IEEE 488, bude připojen pomocí RS232.

Výchozí tovární parametry rozhraní jsou 1200 baudů, parita není, 1start bit, 8 datových bitů (MSB je 0) a 1 stop bit, handshake hardwarový, mód výstupních dat je nastaven pro počítač. Možnosti nastavení jsou uvedeny v příloze č. 6.

Parametry se dají pouze nastavovat pomocí mechanických přepínačů na tlakoměru. Pro hardwarový handshake využívá RTS/CTS piny a v případě, že data jsou připravena, zařízení dává 12 V na RTS. Následně čeká na hodnotu CTS, která musí být větší než 3 V.

Rozhraní na straně tlakoměru je vyvedeno do samice 25pinového konektoru D-submini (DB25S). Originální kabeláž od výrobce nebyla součástí dodávky. Obrázek 5.2 zobrazuje doporučené zapojení kabeláže, podle kterého bude kabeláž zhotovena v rámci implementace.

5.2.3 Popis komunikačního protokolu

Formát zprávy na požadavek zaslání aktuální hodnoty tlaku v režimu PC, je jeden znak a to Carriage Return. Následně tlakoměr zašle odpověď ve formátu zobrazeného v tabulce č. 7.

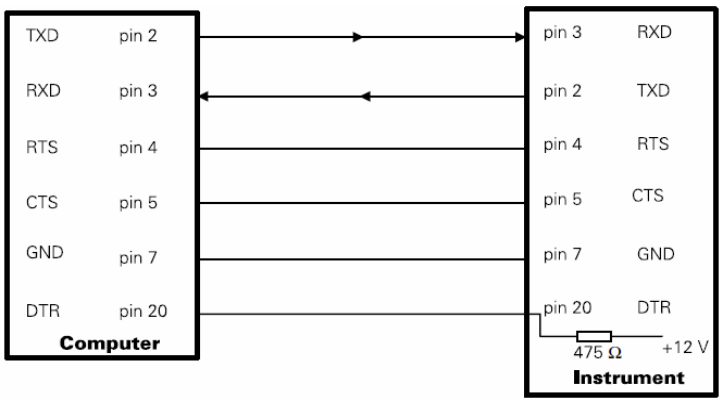
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Z	Č	Č	Č	Č	Č	Č	CR

Tabulka 7: Formát výstupních dat

Popis tabulky:

Z- znaménko, Č – číslice nebo desetinná tečka, CR – Carriage Return

V režimu tiskárny se automaticky data odesílají pětkrát za vteřinu. Mají stejný formát jako data v režimu PC, ale mají na konci o znak více a tímto znakem je Line Feed. Maximální počet hodnot, které je tlakoměr schopen zaslat v režimu PC jsou čtyři za sekundu.



Obrázek 5.2: Zapojení konektorů kabelu k tlakoměru Druck DPI 141 [4]

5.3 Elektronický teploměr Greisinger GMH3710

Při kalibraci se tento teploměr využívá pro měření teploty vážené kapaliny a následně pro výpočet objemu s korekcí na dilataci kapaliny.

5.3.1 Popis zařízení

Základní údaje o teploměru GMH3710 od výrobce GREISINGER electronic GmbH, SRN s výrobním číslem 1341911. Využívá čtyřvodičovou odporovou sondu Pt100. Měřící rozsah přístroje je od -199,99 °C až do 199,99 °C s přesností 0,01 °C, resp. -200.0 °C až 850,0 °C s 0,1 °C, v závislosti na použité sondě. Rozhraní analogové 0 V - 1 V a galvanicky oddělený převodník USB3100 s rozhraním USB.

Návod k převodníku USB3100 je velmi stručný. Při instalaci bylo zjištěno, že převodník využívá čip z řady CP210x firmy Silicon Labs k převodu USB na UART, který se po nainstalování příslušných ovladačů jeví jako virtuální sériový port. Přehled parametrů je v příloze č. 7.

Popis jak se zařízením komunikovat není. Je zde ale možnost zakoupit proprietární knihovnu zapouzdřující komunikaci.

5.4 Elektronický vlhkoměr s teploměrem COMMETER D3121

Hodnoty z tohoto přístroje se používají při výpočtu vztlaku vzduchu. Tyto hodnoty jsou relativní vlhkost vzduchu a teplota vzduchu.

5.4.1 Popis zařízení

Kombinovaný datalogger COMMETER D3121 vyrobila společnosti COMET SYSTEM s.r.o., ČR. Teplotní sonda je odporový snímač Ni1000/6180 ppm s rozsahem měření teploty -30,0 °C až 105,0 °C, při rozlišení 0,1 °C a přesnosti ±0,4 °C. Rozsah měření vlhkosti je v rozsahu 0 až 100 % RV s rozlišením 0,1 % RV a přesností ±2,5 % RV v rozsahu 5 až 95 % RV při 23 °C. Rozhraní je dostupné v podobě RS232C.

Tento přístroj je pro tuto aplikaci nevyhovující z toho důvodu, že se jedná o datalogger a při připojení k rozhraní PC se automaticky přepíná do režimu, ve kterém nelze odečítat aktuální hodnoty. Byl tedy navržen k zakoupení jiný produkt, který má obdobné parametry a lze s ním jednoduše komunikovat při odečítání aktuálních hodnot.

5.5 Elektronický vlhkoměr s teploměrem COMMETER T3311

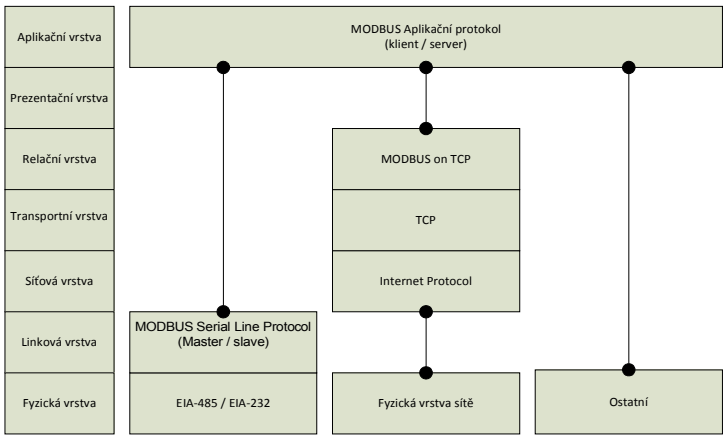
Tento přístroj byl pořízen, aby nahradil pro tento účel nevyhovující datalogger COMMETER D3121.

5.5.1 Popis zařízení

Průmyslový snímač COMMETER T3311 je od stejného výrobce, jako zmíněný nevhodný datalogger a to od COMET SYSTEM s.r.o., ČR. Teplotní sonda je odporový snímač Ni1000/6180ppm s rozsahem měření teploty -30,0 °C až 80,0 °C, při rozlišení: 0,1 °C a přesností ±0,4 °C. Rozsah měření vlhkosti je v rozsahu 0 až 100 % RV s rozlišením 0,1 % RV a přesností ±2,5 % RV v rozsahu 5 až 95 % RV při 23 °C. Pro datovou komunikaci má snímač rozhraní RS232C. Výchozí nastavení je 9600 Bd, bez parity, 2 stop bity. Tento přístroj implementuje nad rozhraním tři komunikační protokoly, které jsou volitelné. Jsou to protokoly Modbus, ADAM společnosti Advantech a ARION od spol. AMIT. Pro implementaci jsem zvolil Modbus, jelikož je nastaven jako výchozí protokol a jedná se o nejvíce rozšířený protokol. Další výhodou je, že existují i open source knihovny, které implementují tento protokol. Níže je zestručněný popis tohoto protokolu.

5.5.2 Popis protokolu Modbus

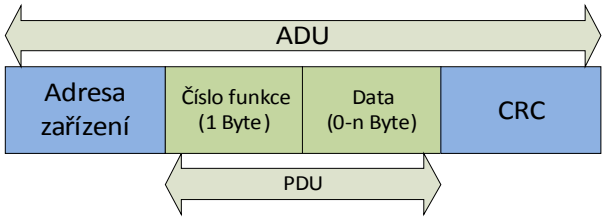
Modbus je otevřený protokol pro přenos diskrétních a analogových dat, který byl vyvinut společností *Gould Medicon* (dnes *Schneider Electric*) v roce 1979, pro komunikaci s *PLC (Programmable Logic Controllers)*. V současnosti tento protokol spravuje organizace *Modbus Organisation*, která se stará o jeho vývoj a aktualizace. Tento protokol lze klasifikovat dle ISO/OSI,



Obrázek 5.3: Zobrazení vrstev protokolu Modbus, informace převzaty z [25,26]

jako bezstavový protokol aplikační vrstvy, který je založený na transakcích. Je hojně rozšířený v průmyslové a řídící technice a je de facto v průmyslu standardem. Protokol poskytuje komunikaci typu klient/server či master/slave na různých typech sběrnic a sítí. U zařízení, které využívají Modbus, je nejrozšířenější sběrnicí RS-485. Další běžnou sběrnicí je RS-232 a také sítě Ethernet. Tyto tři různé sběrnice jsou v aplikacích nejběžnější, ale jak již bylo zmíněno, Modbus není závislý na fyzické vrstvě a lze nalézt implementace i pro jiné typy sběrnic a sítí. Díky této vlastnosti, lze za pomoci bran vytvořit architekturu, ve které jsou různé typy sběrnic, ale všechny zařízení komunikují stejným protokolem [23,24].

Transakce jsou založeny na požadavcích a odpovědích. Transakce s požadavky zasílá klient a odpovídá na ně server. Transakce využívají jednoduchých paketů – *PDU (Protocol Data Unit*), které jsou nezávislé na nižších vrstvách. PDU obsahuje dvě pole – první je o velikosti jeden bajt obsahující kód požadované funkce a pole dat, které má proměnlivou velikost, které zde nemusí být vůbec, v



Obrázek 5.4: Znázornění datového rámce

závislosti na sémantice požadované funkce. Specifikace uvádí tři typy PDU - prvním typem je Modbus *Request PDU*, obsahující na prvním místě PDU bajt nesoucí kód požadované funkce a *N* bajtů dat pro tuto funkci (např. specifikace počtu požadovaných bajtů, adresy apod.). Druhým je Modbus *Response PDU,* obsahující bajt s kódem odpovídající požadavku a *N* bajtů s hodnotou odpovědi a poslední Modbus *Exception PDU* obsahuje bajt s hodnotou kódu požadované funkce s přičtenou hodnotou 0x80h a pak následuje druhý bajt nesoucí specifický kód chybové výjimky [24,25].

Kódy funkcí, které určují požadovanou činnost po zařízení typu slave/server jsou v rozsahu 1 – 255, Tento prostor je rozdělený podle účelu do pěti podprostorů. Rozsah 1 – 65, 73 - 100 a 111 - 127 je vyhrazený pro veřejné funkční kódy, 66 – 72 a 101 - 110 pro uživatelsky definované kódy. Veřejné funkční kódy jsou unikátní, definované, validované Modbus organizací a mají uveřejněnou dokumentaci funkce. To znamená, že pokud zařízení implementuje nějaký takovýto kód, musí být v souladu s touto dokumentací. Uživatelsky definované kódy slouží k implementaci funkcí, které nejsou uvedeny ve specifikaci. Hodnoty 128 – 255 jsou vyhrazeny pro výjimky a kód 0 je nevalidní. Seznam kódů je popsán v [25] a níže se budu zabývat popisem funkcí, které jsou implementovány v T3311.

Pro navázání aplikační vrstvy na konkrétní sběrnici se využívá *ADU (Application Data Unit)*, která přidává k PDU další pole, které specifikuje adresu zařízení a kontrolní součet, viz obrázek 5.3. Pro sériové rozhraní má ADU před PDU pole s adresou o velikosti jednoho bytu a za PDU dva byty CRC16, tak jak je zobrazeno na obrázku č. 5.4. Maximální délka zprávy je závislá na konkrétní sběrnici. Pro IEEE 485 je maximální velikost ADU 256 B a po odečtení adresy a CRC zbývá 253 B pro PDU [25].

Datový model tohoto protokolu definuje čtyři základní datové typy, která jsou uvedeny v tabulce.

Název dat	Velikost [bit]	Typ přístupu	Popis
Diskrétní vstup	1	Čtení	Tento typ dat mohou poskytovat I/O systémy
Cívky	1	Čtení/Zápis	Tento typ dat může být měněný aplikačním sw.
Vstupní registry	16	Čtení	Tento typ dat mohou poskytovat I/O systémy
Vlastní registry	16	Čtení/Zápis	Tento typ dat může být měněný aplikačním sw.

Tabulka 8: Modbus - Tabulka základních datových typů. Převzato z [25]

Z přehledu poněkud vybočuje neobvyklým názvem typ, zvaný cívka. Tento typ může například reprezentovat fyzické relé.

Specifikace přesně stanovuje adresování v PDU a to tak, že v PDU jsou všechna data adresována od 0 do 65535. Také specifikuje, že každý element v datovém bloku je adresován od 1 do n. Data jsou kódována do Big Endian [25].

Organizace Modbus vydala také specifikace implementace pro nejrozšířenější typy sběrnic. Dále se budu pouze zabývat specifikací implementace pro sériová rozhraní, kterou implementuje T3311. Tato implementace může využívat dvě rozhraní – rozšířenější IEEE 485 a RS-232, proto jsou uvedené informace obecné a počítají se sběrnicí *point to multipoint*, což je IEEE 485. Pro RS-232, která je typu *point to point*, se situace v oblasti adresování zjednodušuje. Komunikace probíhá způsobem, že je ke sběrnici připojeno jedno zařízení typu master a až 247 zařízení typu slave. Transakce je vždy iniciovaná zařízením typu master a zařízení typu slave nesmí vyslat data, aniž by bylo dotázáno [26].

Adresový prostor je 0 až 255 a je rozdělen následovně. Adresa 0 je vyhrazena pro *broadcast*, na který zařízení typu slave neodpovídá a je určen k zapsání např. parametrů všem zařízením typu slave. Specifikace také uvádí, že všechna zařízení musí rozpoznávat *broadcast* [26].

Adresy 1 – 247 jsou vyhrazeny pro *unicast* a každému zařízení typu slave náleží jedna unikátní adresa z tohoto rozsahu. Zařízení typu master se neadresuje a proto nemá adresu. Transakce se pak skládá ze dvou zpráv – požadavku od zařízení master a odpovědi od zařízení typu slave. Adresy 248 až 255 jsou vyhrazené [26]. Snímač Comet T3311 má z výroby nastavenou adresu 1 a vzhledem k tomu, že má RS-232, je nepravděpodobné, že by se adresa někdy měnila.

ADU pro sériovou komunikaci odpovídá obrázku 5.4. Poslední pole na obrázku 5.4 je vyhrazeno pro kontrolní součet, který může být buď *CRC*, nebo *LRC (Longitudinal redundancy check)* v závislosti na zvoleném přenosovém módu [26].

Módy jsou dva a to *Modbus RTU (Remote Terminal Unit),* a nebo *Modbus ASCII.* Zvolený mód musí být u všech zařízení, připojených na sběrnici, stejný. Navíc dle specifikace musí všechna zařízení implementovat Modbus RTU a zároveň by měl být výchozím typem přenosu [26].

Modbus RTU, který je implementován ve snímači T3311, je výhodný z hlediska datové propustnosti – jeden bajt nese dva čtyřbitové hexadecimální znaky a tím získává oproti Modbus ASCII větší datovou hustotu. Každá zpráva musí být odeslaná ve stanoveném kontinuálním proudu bitů. Ten je dán takto: 1start bit, 8 datových bitů, 1 paritní bit a 1 stop bit. Celkem je tedy odesláno 11 bitů. Dále je specifikováno, že výchozí parita musí být sudá. Pokud se bude využívat bezparitní přenos, je nutné použít dva stop bity [26].

Zařízení odesílá data v podobě rámců. Tyto rámce mají ve specifikaci definovaný časový rozestup, který je větší nebo roven době přenosu 3,5 znaků (bitů). Při odesílání rámce nesmí vzniknout mezi jednotlivými znaky rámce větší prodleva než 1,5 znaku. V případě delší doby, specifikace definuje chybu přenosu [26].

Modbus RTU využívá *CRC (Cyclical Redundancy Checking)* pro kontrolu integrity přenesených dat. Využívá se 16bitové CRC, označované jako *CRC16-IBM*, využívající polynom $x^{16}+x^{15}+x^2+1$ tj. *OxA001h* [26]. Algoritmus pro výpočet CRC16 je v příloze č. 2.

Protokol *Modbus ASCII* využívá k přenosu dat 7bitové ASCII kódování. Každá 8bitová zpráva je přenášena dvěma 7bitovými ASCII znaky. Např. hodnota 0x5B se přenese jako 0x35(5) a 0x42(B). Z toho plyne nevýhoda tohoto přenosu a to nižší datová propustnost oproti Modbus RTU. Místo CRC se využívá LRC [26]. Díky faktu, že použité přístroje tuto variantu nepodporují, dále se touto variantou nebudu zabývat.

V návaznosti na část o funkcích, bude zde stručně popsáno, jaké funkce má implementované snímač Comet T3311. Jsou to funkce č. 03 (0x03) - „*Čtení 16bitových registrů*“ (*Read Holding Registers*) a slouží k vyčtení hodnot ze zařízení. Jako parametry se předávají počáteční adresa registru a počet registrů k vyčtení. Funkce č. 04 (0x04) s názvem „*Čtení 16bitové vstupní brány*“ (*Read Input Registers*) má stejnou syntaxi i sémantiku, jako u předchozí funkce. Poslední funkcí je funkce č. 16 (0x10) s názvem „*Nastavení více 16bitových registrů*“ (*Write Multiple Registers*) a slouží k zápisu do registrů zařízení. Lze tak změnit nastavení snímače, jako jsou např. jednotky, nastavení komunikace atd. [27].

6 Návrh a implementace aplikace

V první části se budu zabývat popisem zjišťování požadavků na vytvářený produkt a důkladné seznámení s cílem. Dále z těchto požadavků vytvořím návrh aplikace, ve formě diagramu tříd. Před implementací jsem si také ověřil výše popsanou komunikaci s jednotlivými přístroji. Díky tomu jsem získal surová data, ze zařízení, které využiji pro pozdější implementaci emulátorů těchto přístrojů. V této fázi bylo také nutné vytvořit chybějící potřebnou kabeláž dle doporučeného zapojení od výrobce. Na základě těchto testů jsem se rozhodl, že se pokusím vytvořit vlastní implementaci komerční knihovny, která se prodává k teploměru Greisinger. Pro tuto implementaci bylo nutné použít odposlech komunikace, a proto je zde celý postup analýzy popsán.

6.1 Specifikace systémových požadavků

6.1.1 Zjišťování požadavků na pracovišti a přímé pozorování procesu

Pozorováním a praktickým vyzkoušením celého procesu v laboratoři jsem podrobněji zjišťoval úkony a údaje, které se provádějí a využívají při kalibraci pístových pipet. Pipeta a její parametry jsou důležitou entitou, kterou bude potřeba v systému modelovat. Parametry pipety určují počet měření a jsou také nutné pro výpočet. Je tedy potřeba, aby uživatel před zahájením měření, zvolil typ pipety. Může zvolit, zda je pipeta jedno, nebo vícekanálová a typ objemu. Pipety mohou mít fixní, nebo nastavitelný objem. V případě zvolení pipety s nastavitelným objemem, je nutné nastavit hodnoty objemů, ve kterých bude probíhat měření. Norma doporučuje měření pro tři objemy na každý kanál. V případě fixní pipety bude nastavena nominální hodnota objemu této pipety. Při kalibraci se měření hodnoty každého zvoleného objemu opakuje n-krát, přičemž norma stanovuje *n=10*. Z důvodu flexibility, by bylo vhodné umožnit tuto hodnotu také libovolně nastavovat. Celkově je tedy potřeba nasnímat a uložit *K* hodnot, přičemž *K* lze stanovit takto: *K = počet kanálů * počet nastavených objemů * n opakování měření*. Do budoucna by aplikace mohla obsahovat jednoduchou databázi typů pipet a to z důvodu urychlení celého procesu a odstranění možnosti chyb při zadávání těchto hodnot parametrů pipet.

Každá hodnota (iterace měření) bude obsahovat hodnoty: váha roztoku, teplota okolí, vlhkost okolí a atmosférický tlak. Vzhledem k tomu, že u vícekanálových pipet se může jednat o velké množství měření, bylo by vhodné umožnit rozpracované měření uložit a načíst pro následné doměření.

Z důvodu možnosti zanesení náhodné chyby, bude možné hodnoty zkontrolovat a případně nekorektní hodnoty znovu změřit, nebo smazat. V další verzi aplikace by bylo možné zvážit možnosti automatické detekce těchto odlehlých hodnot. Vzhledem k zažitému systému ukládání hodnot do sešitu v aplikaci MS Excel, kde se následně i provádí výpočet, budou následně hodnoty exportovány do připraveného sešitu aplikace MS Excel. Tento sešit bude dopředu připravený a pro každou kalibraci se zkopíruje. To bude mít výhodu i v případě nutnosti změny ve výpočtu. Název nového souboru se bude skládat z aktuálního data a čísla zakázky. Číslo zakázky bude uloženo i v sešitu, a proto bude také zadáno před spuštění měření. V budoucnu by aplikace mohla obsahovat databázi zákazníku a jejich měřidel.

Vzorový sešit bude obsahovat jeden list s výpočty. Tento list se bude kopírovat pro každý měřený objem. Do tohoto listu se pak uloží jednotlivé hodnoty opakovaného měření. V případě vícekanálové pipety bude název listu obsahovat číslo kanálu. Eventuálně by aplikace mohla i obsahovat jednoduchou databázi zakázek, zákazníků a jejich měřidel. Taktéž by mohla obsahovat role, které by zajistily nemožnost manipulace s daty neoprávněnými osobami. Role by mohly být následující – správce, vedoucí laboratoře a laborant. Správce by měl možnost nastavovat komunikaci s přístroji, nastavení výstupu a také by měl možnost vytvářet nové uživatele libovolné role. Vedoucí pracovník by měl možnost úplné manipulace s DB, včetně mazání a editace záznamů. Také by mohl měnit a vytvářet uživatele systému a provádět měření.

Z tohoto intuitivního popisu doplněného konzultacemi, jsem sestavil semiformální požadavky. Jsou to funkční a nefunkční požadavky, tak jak je doporučeno v knize [28].

6.1.2 Slovníček pojmů

Dle doporučení v [28], zde uvádím pojmy, které jsou často užívané využívané v projektu, zejména jeho popisu (z důvodu vyhnutí se dezinterpretace pojmů):

- *měření* – „je proces experimentálního získávání jedné, nebo více hodnot veličiny, které mohou být důvodně přisouzeny veličině“[31]
- *postup měření* – „je podrobný popis měření, podle jednoho nebo více měřících principů a dané metody měření, založený na modelu měření a zahrnující jakýkoliv výpočet k získání výsledků.“[31]
- *měřící systém* – „je soubor jednoho nebo více měřidel a často dalších zařízení, včetně jakýchkoliv chemikálií a napájení, sestavených a přizpůsobených k poskytování informace používané ke generování naměřených hodnot veličin ve specifikovaných intervalech, pro veličiny specifikovaných druhů.“[31]
- *pipeta* – „měřidlo pro přesné odměřování vylitých objemů“[32]
- *pístová pipeta* – „je přístroj, který využívá pístového válce k odměření objemu“[32]

- *kalibrace pipet* – „je normou stanovená metodika pro určování chyby odměřovaných objemů“[32]

6.1.3 Funkční požadavky

1. Měřící aplikace bude čekat na hodnotu z vah [M]
2. Měřící aplikace bude po obdržení hodnoty z vah odečítat hodnoty z ostatních měřidel [M]
3. Měřící aplikace bude zobrazovat odečtené hodnoty [S]
4. Měřící aplikace bude umožňovat znovu změřit vybrané hodnoty [S]
5. Měřící aplikace bude umožňovat export hodnot do sešitu aplikace MS Excel [M]
6. Měřící aplikace bude schopna uložit resp. načíst rozpracované měření [M]
7. Měřící aplikace bude umožňovat nastavit parametry pipety [M]
8. Měřící aplikace bude z parametrů pipety odvozovat měřící cyklus [M]
9. Měřící aplikace bude uchovávat a spravovat informace o jednotlivých typech pipet [C]
10. Měřící aplikace bude uchovávat a spravovat informace o zákaznících [W]
11. Měřící aplikace bude implementovat roli správce [W]
12. Měřící aplikace bude implementovat roli vedoucího pracovníka [W]
13. Měřící aplikace bude implementovat roli laboranta [W]

6.1.4 Nefunkční požadavky

1. Měřící aplikace bude desktopová aplikace s GUI
2. Měřící aplikace bude spustitelná v MS Windows XP SP3 a novější
3. Měřící aplikace bude využívat stávající přístroje v laboratoři

Atributy požadavků: M – nezbytný (Must have) - povinné požadavky, jež jsou základem systému, S – možný (Should have) – důležité požadavky, které lze však vynechat, C – Eventuální (Could have) – požadavky, jež jsou opravdu nepovinné, W – Chceme mít (Want to have) – požadavky, které mohou být zahrnuty do dalších verzí systému [27].

6.1.5 Případy užití

V případu užití vystupují tito aktéři:

- *Laborant* – tento aktér realizuje měřící postup, který počíná fází zjišťování parametrů pipety, ze kterých se stanoví celkový počet iterací měření. Pokud tento typ pipety byl již ověřován, je možné jej vyhledat v databázi parametrů pipet. Pokud databáze neobsahuje přednastavený záznam, může laborant zadat nový typ pipety do databáze, včetně údajů o výrobci. Může také opravovat špatně zadané údaje a mazat již nepoužívané typy pipety. Po provedení měření, může v aplikaci zkontrolovat naměřené hodnoty, a pokud je potřeba, může vybrané odlehlé

hodnoty znovu změřit. Poté následuje export hodnot do sešitu aplikace MS Excel. Pokud je potřeba, může měření uložit a později načíst a pokračovat od poslední iterace měření. Tato situace může nastat například při měření, které je obsáhlé a časově náročné – vícekanálové pipety.

- *Vedoucí pracovník* – může vykonávat stejné činnosti jako laborant, ale navíc disponuje pravomocemi pro zadávání a editaci zakázek do databáze, včetně potvrzování výsledků, pokud je to nutné. V případě nových neevidovaných zákazníků, má možnost zadávat údaje o zákaznících. Také má možnost tyto údaje editovat a rušit. Posledním případem užití aplikace je, že může přidávat nové uživatele do systému, jako jsou např. další laboranti.
- *Správce* – má pravomoce k nastavování aplikace a měřící zařízení, které tato aplikace využívá. Také má pravomoc pro vytváření nových uživatelů systému a přiřazování jejich rolí.

6.1.6 Diagram případu užití

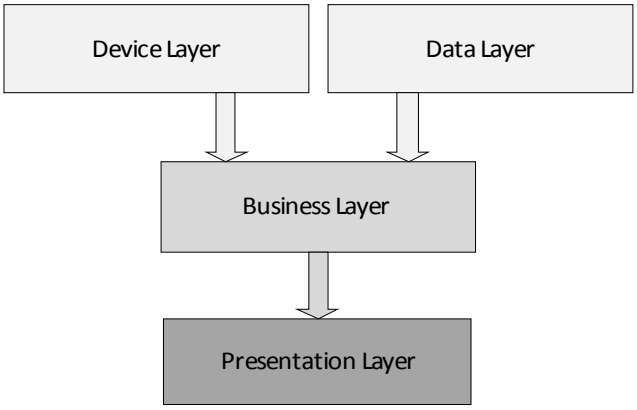
Obrázek 6.1: Diagram případu užití, zahrnující všechny atributy funkčních požadavků

6.1.7 Specifikace případu užití

Případ užití: Kalibrace měřidla
ID: KalM 1
Účastníci: Vedoucí laboratoře Laborant
Vstupní podmínky: Přijetí měřidla(pipety) ke kalibraci Všechny využívané měřicí přístroje jsou připojeny a jsou v pořádku
Tok událostí: 1. Případ začíná vyhledáním zákazníka, vedoucím laboratoře 2. KDYŽ není zákazník evidovaný 1. Vedoucí laboratoře přidá zákazníka do evidence 3. Vedoucí laboratoře zadá zakázku 4. KDYŽ typ pipety není evidovaný 1. Vedoucí laboratoře přidá pipetu do aplikace 5. Laborant zahájí měření 6. Laborant překontroluje naměřené hodnoty 7. Laborant exportuje výsledky do sešitu aplikace MS Excel
Následné podmínky: V sešitu proběhne potřebný výpočet a hodnoty jsou dosazeny do kalibračního listu

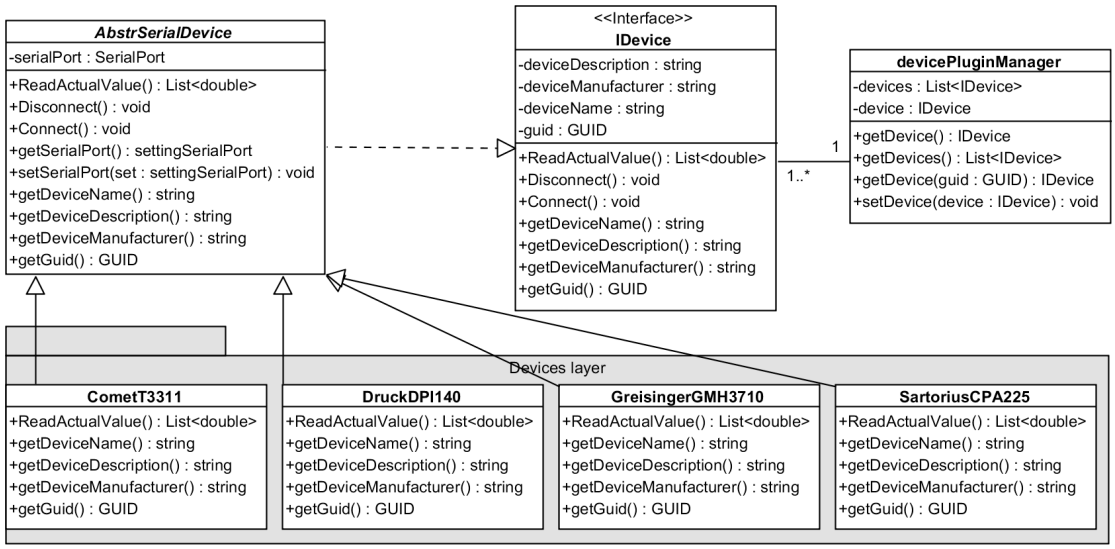
6.1.8 Vrstvená architektura

Aplikaci jsem rozdělil do více vrstev, které by měly umožnit snazší údržbu v budoucnu. Konceptuální návrh aplikace jsem dekomponoval do čtyř vrstev, které budou popsány níže. Celý koncept včetně tříd je v příloze č. 11.



Obrázek 6.2: Návrh vrstvené architektury

Vrstva *Device layer* reprezentuje všechny měřicí přístroje. Každý přístroj je reprezentován vlastní *dll* knihovnou, která implementuje rozhraní IDevice. Tyto knihovny se dynamicky načítají v *Business layer* pomocí tzv. *plugin frameworku*. Diagram popisující tuto vrstvu a plugin framework je na obrázku č. 6.3.



Obrázek 6.3: Datová vrstva a plugin framework, který se nachází v Business Layer

Vrstva *Data Layer* obsahuje třídy, které reprezentují jednotlivé databázové tabulky. Pro manipulaci s touto vrstvou jsou ve vrstvě *Business Layer* navrhnuty třídy, které zajišťují veškerou manipulaci s těmito třídami, včetně synchronizace s databází.

Vrstva *Business Layer* je vrstvou, ve které se soustředí jádro aplikace. Je v ní již zmíněný plugin framework, třída zapouzdřující práci s aplikací MS Excel, která implementuje rozhraní IspreadsheetApp. Toto rozhraní je zde z důvodu možnosti rozšíření implementace využívající jiný tabulkový procesor. Dále jsou zde třídy na správu databázových dat a řízení měření.

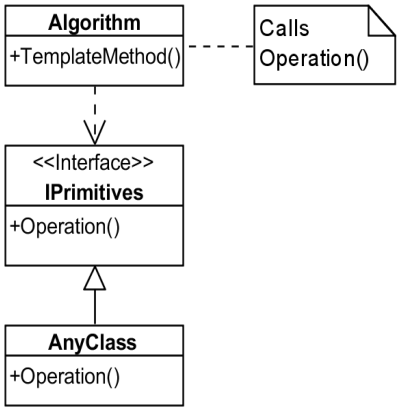
Vrstva *Presentation Layer* reprezentuje uživatelské rozhraní. V konceptu tato vrstva není úplně dokončená. Třídy, které obsahuje, reprezentují jednotlivé widgety grafického rozhraní, popř. s velmi omezenou logikou v jejich metodách. Je to z důvodu zachování rozdělení do příslušných vrstev.

6.1.9 Použité návrhové vzory

Template method – tento návrhový vzor spadá do kategorie návrhových vzorů pro chování. Definuje kostru algoritmu, který má určité svoje části umístěné v podtřídách. Tedy struktura algoritmu se nemění, ale specifické části jsou uložené v podtřídách [29, 30]. Tento návrhový vzor jsem primárně využil v plugin frameworku.

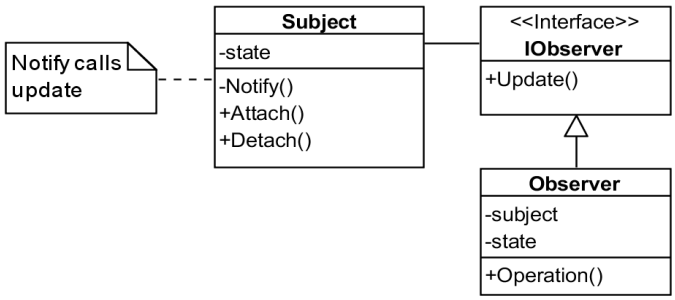
Popis obrázku 6.4: „*Algorithm* – třída, která zahrnuje TemplateMethod, *TemplateMethod* – metoda, která odkládá nějaké části své operace do jiných tříd, *IPrimitives* – rozhraní definující operaci(e), které TemplateMethod odkládá do jiných tříd, *AnyClass* – libovolná třída, která

implementuje rozhraní IPrimitives, *Operation* – jedna z metod, kterou TemplateMethod potřebuje pro dokončení své operace. " [29].



Obrázek 6.4: UML diagram tříd Template Method. Převzato z [29].

Observer je návrhový vzor kategorizovaný do vzorů chování. Tento vzor definuje vztah mezi objekty takového typu, že pokud nastane změna u pozorovaného objektu, je odeslána zpráva o změně objektům, kteří se přihlásili k pozorování tohoto objektu [29,30]. V jazyce C# je tento návrhový vzor využitý u implementace signálů.



Obrázek 6.5: UML diagram tříd: Observer. Převzato z [29].

Popis obrázku 6.5: „ *Subject* – třída jejíž instance mění nezávisle svůj stav a informuje objekty Observer, *IObserver* – rozhraní pro Observer specifikující, jak by měli být pozorovatelé informováni o změnách, *Observer* – třída, která poskytuje metodu Update, aby mohla zůstat konzistentní se stavem Subject, *Update* – operace, která slouží jako rozhraní mezi Subject a Observer, *Notify* – použití mechanismu událostí pro volání operací Update na všech objektech Observer." [29].

6.2 Příprava implementace - testování komunikace

Pro otestování komunikace s přístroji, která byla popsána v návodech, byl zvolen program RealTerm: Serial Capture Port 2.0.0.57 pro jeho bohaté nastavení a schopnost zobrazovat přijatá data

v mnoha formátech. Při testování bylo cíleně bráno výchozí nastavení. Důvod je takový, že pokud by došlo například k servisnímu zákroku, při kterém by bylo zařízení vy-resetováno do továrního nastavení, tak aby bylo možné po návratu ze servisu aplikaci bezproblémově používat dál s výchozím nastavením. Druhým důvodem je skutečnost, že všechna zařízení jsou v tomto nastavení již několik let provozovány a tudíž by změna mohla mít spíše negativní projevy.

6.2.1 Váhy Sartorius CPA225D

Komunikací s váhami byl nasbírán dostatečný počet vzorků dat, které byly následně využity při emulaci vah. Byla testovaná pouze varianta, kdy se data odesílají pomocí tlačítka na vahách.

Komunikace s váhami (nastavení komunikace: 1200bd, 7N1, lichá, RTS/CTS):

N + 0.00001 g CRLF

4E 20 20 20 20 20 2D 20 20 30 2E 30 30 30 30 31 20 67 20 20 0D 0A

N - 0.00001 g

4E 20 20 20 20 20 2D 20 20 30 2E 30 30 30 30 31 20 67 20 20 0D 0A

N +200.0030 g

4E 20 20 20 20 20 2B 32 30 30 2E 30 30 33 30 20 20 67 20 20 0D 0A

Stat H

53 74 61 74 20 20 20 20 20 20 20 20 48 20 20 20 20 20 20 20 0D 0A

Získané hodnoty byly ověřovány s laboratorními závažími a hodnotou na displeji vah.

6.2.2 Tlakoměr Druck DPI 141

Komunikace s tlakoměrem proběhla dle předpokladů a bez jakýchkoliv potíží.

Komunikace s tlakoměrem (4800bd, 8N1, žádná):

Odesláno: CR

Přijato: +969.448 CR (2B 39 36 39 2E 34 34 38 20 0D)

Získané hodnoty byly porovnány s hodnotou na displeji přístroje.

6.2.3 Teploměr s vlhkoměrem COMMETER D3121

Bylo zjištěno, že při připojení k sériovému portu se přístroj automaticky přepíná do režimu 'DATA', ve kterém nenačítá hodnoty (pravděpodobně z důvodů zamezení vzniku nekonzistentních dat) a lze pouze načítat hodnoty z interní paměti a ani výrobce neudává žádný postup, jak komunikovat s tímto přístrojem.

6.2.4 Teploměr GREISINGER GMH3710

Výrobce nikde neuvádí, jak lze s přístrojem komunikovat z externí aplikace. K převodníku USB3100, ale dodal aplikaci, která je schopna zobrazit aktuální hodnotu.

6.3 Řešení komunikace s problematickými přístroji

6.3.1 Teploměr-vlhkoměr COMMETER D3121

Tento teploměr-vlhkoměr se i po konzultaci s výrobcem ukázal pro tento účel jako nevyhovující. Po prodiskutování problému se zadavatelem jsem navrhl zakoupení výše uvedeného snímače COMET T3311, který bude dodán v brzké budoucnosti.

6.3.2 Odposlech a analýza komunikace mezi Greisinger GMH 3710 a dodaným SW

Řešení tohoto problému spočívalo v odposlechu a zanalyzování komunikace mezi teploměrem a konfigurační aplikací GMHKonfig, která byla dodána s převodníkem USB3100. Pro odposlech komunikace byl využit program Free Serial Port Monitor v. 3.31 od společnosti HHD Software Ltd. a také již výše zmíněný RealTerm.

Postup analýzy spočíval v hledání změn v přijímaných datech z teploměru. Značně zkrácený, ale dostatečně velký vzorek dat na vypozorování změn v datech se nachází v příloze č. 4.

Nejprve bylo zjištěno, které sekvence jsou periodické a neměnné. Po zjištění, která část dat se mění v závislosti na teplotě, bylo nutné zjistit, jak jsou hodnoty teploty reprezentovány. Klíčovou částí této analýzy bylo vytvoření trojného bodu, aby mohlo být zjištěno, jak je reprezentovaná 0.00 °C. Pro vystihnutí okamžiku dosažení nuly na obou desetinných místech, bylo třeba experimentům věnovat dostatek času, jelikož i drobná změna polohy teplotní sondy rozvířila ledovou tříšť a následným skokem teploty v setinách °C. Dále bylo zapotřebí zjistit, jak se reprezentují záporné hodnoty.

Tabulka č. 7 obsahuje několik extrahovaných zpráv, přijatých od teploměru a zároveň odečtená hodnota z displeje teploměru. Z tohoto vzorku je patrné, že změny hodnoty teploty ovlivňují poslední tři bajty. Dále je také patrné, jak jsou reprezentované záporné hodnoty. Postupnou analýzou bylo zjištěno, že klíčové jsou dva předposlední bajty. Poslední bajt se mi nepodařilo identifikovat, ale je pravděpodobné, že se jedná o kontrolní součet.

Teplota odečtená z displeje teploměru	Data odeslaná teploměrem
23,81 °C	FE 05 26 71 00 48 F6 4D 71
27,28 °C	FE 05 26 71 00 48 F5 A8 FB
26,55 °C	FE 05 26 71 00 48 F5 5F 30
-00.03 °C	FE 05 26 72 FF 84 00 FD 02
-00.02 °C	FE 05 26 72 FF 84 00 FE 0B
00.00 °C	FE 05 26 71 00 48 FF 00 28
00.10 °C	FE 05 26 71 00 48 FF 0A 1E
42.73 °C	FE 05 26 71 00 48 EF B1 61

Tabulka 9: Vzorek dat pro analyzování hodnot

Z těchto dvou předposledních hodnot jsem odvodil vztah pro převod na číslo s desetinnou čárkou *číslo = (256 * (neg(A)) + B) / 100*, kde A je sedmý bajt a B je osmý bajt.

Dále byla testována různá nastavení teploměru, jako je například změna jednotky, nastavení offsetu atd. a kompletní výsledky jsou přiloženy na CD. Důležitým poznatkem je, že pokud je nastaven offset, tak hodnota, která se získá metodou popsanou výše, obsahuje již tento offset započítaný. Tedy dá se zobecnit, že zmíněné bajty, které nesou číselnou hodnotu teploty, jsou vždy totožné s údajem na displeji teploměru, který zobrazuje korigovanou hodnotu.

Následovaly experimenty jak programově vyvolávat aktuální hodnoty a došlo se k závěru, že stačí odeslat tři bajty: 0xFE, 0x00, 0x3D. Tento způsob byl podrobně testován a nebylo dosaženo žádného nekorektního výsledku.

6.4 Implementace aplikace

Pro vývoj byl zvolen jazyk C#, který byl vyvinut společně s .NET Frameworkem. Zvolil jsem jej pro jeho vysokou úroveň abstrakce a provázanost s .NET Frameworkem. Tento framework od verze 2.0 obsahuje třídu SerialPort, která umožňuje práci se sériovým portem na vysoké úrovni abstrakce. Vzhledem k nefunkčnímu požadavku č. 2, „Měřící aplikace bude spustitelná v MS Windows XP SP3 a novější“, lze použít aktuální verzi - .NET Framework 4.0.

V první fázi, z důvodu vyzkoušení si práce s tímto prostředím a také z praktického hlediska, byly nejprve vytvořeny „emulátory“ vah a tlakoměru. Jejich princip byl velmi zjednodušený a realizoval pouze odesílání dat a přijímání dat. Data, která jsou odesílána emulátory, jsou data, jež byla získána při testování komunikace s přístroji. Komunikace mezi vyvíjenou aplikací a emulátory probíhala přes virtuální sériové porty, vytvořené pomocí aplikace *Null-modem emulator (com0com v2.2.2.0)*.

Emulátory byly také vytvořeny z důvodu nemožnosti být při vývoji neustále v laboratoři. Tyto emulátory mají pouze za cíl testování reakcí hlavní aplikace a jsou na velmi vysoké úrovni abstrakce. Neimplementují žádné zpoždění apod., které nastávají při reálné komunikaci.

První verze aplikace, byla oproti výše uvedenému návrhu, zjednodušena. Každé zařízení reprezentuje jedna třída a implementuje metody pro načtení hodnot a následné parsovaní hodnoty. Všechny tyto třídy implementují metodu getValue(), která vrací načtenou hodnotu typu double. Konstruktor očekává parametry portu, ke kterému je zařízení připojeno. Dále byla vytvořena třída, která tato data reprezentuje. Účelem této třídy je, aby její instance reprezentovala hodnoty ze všech přístrojů, a kolekce těchto instancí bude reprezentovat všechny iterace jedné kalibrace. Dále zde je třída abstrahující práci s aplikací MS Excel, která využívá Microsoft Office PIA, jež bude popsáné níže. Pro ukládání nastavení se využívá třída ApplicationSettings z .NET frameworku. Ačkoliv tato verze má určité třídy společné s návrhem, jedná se o prototyp, na kterém jsem se seznamoval s tímto jazykem.

6.4.1 Microsoft Office Primary Interop Assemblies (PIAs)

Pro přístup k aplikaci Excel se využívá jejího objektového modelu. Díky tomuto modelu lze ovládat téměř libovolnou funkci této aplikace. Pro zpřístupnění rozhraní objektového modelu aplikace je nutné ve vývojovém prostředí zavést reference na Microsoft Office PIA, který se s novějšími verzemi aplikace Excel (2003 a novější) instaluje automaticky. Pro starší Excel XP je nutné tyto knihovny stáhnout a doinstalovat. Po přidání referencí se objeví jmenný prostor Microsoft.Office.Interop. V něm se nachází aplikace balíku Office. Po zpřístupnění rozhraní lze jednoduchým způsobem otevřít aplikaci Excel a dále s ní pracovat. Jelikož se ale pracuje s COM objekty je nutné při ukončení práce explicitně vyvolat úklid paměti. Obavy z omezení přenositelnosti aplikace z důvodu různých verzí MS Excelu a jejich knihoven PIAs, se ukázala jako lichá. Základní funkcionalita jako je otevření souboru, uložení souboru, nastavení aktuálního listu, vložení hodnot funguje i v nových verzích(2007) a to za použití PIAs z verze XP. První verze aplikace využívá právě tyto staré knihovny.

7 Závěr

Aplikace pro podporu kalibrace pístových pipet byla vyvinuta za účelem zefektivnění procesu jejich kalibrace. Z krátkodobého experimentování bylo zjištěno, že program ušetří přibližně 7 minut z průměrných 20 minut jednoho měření nejběžnějšího typu pipety. Při průměrné kalibraci osmi pipet denně, vychází úspora téměř jedné „člověko-hodiny“. Tato úspora času se pravděpodobně ještě zvýší, až bude měřící sestava doplněna o nový teploměr s vlhkoměrem, který bude možné zakomponovat do této měřící sestavy. Navíc až bude sestava kompletní, tak se úplně zamezí vzniku náhodné chyby při přepisu hodnot.

Současná verze implementuje nezbytné funkční požadavky, které byly popsány v kapitole 6.1.3. Nicméně z pohledu kvality kódu je znát, že zvolený jazyk C# a hlavně velmi obsáhlý .NET Framework jsem studoval při vypracovávání této práce a jsem si vědom, že tento projekt potřebuje pro jeho budoucí rozvoj refaktorizaci kódu.

Na druhou stranu mi tato práce umožnila získat mnoho nových praktických i teoretických zkušeností. Jelikož jsem si zvolil práci s vlastním zadáním a zadavatel je odborníkem v naprosto jiné oblasti, musel jsem se v této práci podrobněji zabývat zjišťováním požadavků na projekt. Ty jsem zjišťoval přímo na pracovišti diskusí se zadavatelem, pozorováním a praktickým vyzkoušením všech fází kalibrace pístových pipet a také nepřímo analýzou normy, která stanovuje kalibrační metodiku. Na základě těchto zjištěných požadavků jsem se snažil vytvořit návrh aplikace. Od počátku bylo mým cílem vytvořit aplikaci, která bude použitelná i v jiných laboratořích, a proto jsem se snažil navrhnout aplikaci s vrstvenou architekturou. Vzhledem k přenositelnosti jsem zanalyzoval dostupná zařízení a jejich rozhraní včetně komunikačních protokolů, abych mohl vhodně navrhnout jednotlivé vrstvy. Z této fáze vzešla myšlenka vyčlenění implementace popisu zařízení do knihoven, které budou v aplikaci formou pluginů. Návrh tedy obsahuje i plugin framework, který dynamicky přidává do aplikace tyto pluginy.

Před implementací bylo také nutné nastudovat komunikační protokoly jednotlivých měřidel, které jsou používány při kalibraci na ČMI v Brně. Tato fáze vyústila v řešení komunikace se zařízeními, u kterých nebyl dostupný popis komunikačního protokolu. V případě teploměru od společnosti Greisinger se mi podařil problém vyřešit pomocí odposlechu komunikace, na jehož základě jsem vytvořil vlastní knihovnu. Nepodařilo se mi ale vyřešit komunikační problém s dataloggerem od společnosti Comet, který integruje teploměr a vlhkoměr. Jako řešení jsem navrhl průmyslový snímač s ekvivalentními parametry od téže společnosti, který využívá protokol Modbus RTU s volně dostupnou specifikací.

V současnosti se také zabývám otázkou, zda by nebylo vhodné aplikaci rozšířit i o kompletní výpočet kalibrace, který je nyní realizován v tabulkovém procesoru. Zabývám se tím z toho důvodu,

že integrita dat a výpočtů v tabulkovém procesoru může být jednoduše narušena, ať již z nepozornosti, či úmyslně a šíření chyb ve výpočtech je pak velmi snadné, obzvlášť při takto rozsáhlých výpočtech. Jistou variantou řešení této situace je zamykání klíčových buněk. Do budoucna bude nutné tuto otázku vyřešit.

Literatura

[1] ISO: *ČSN EN ISO 8865-6:2003: Pístové titrační přístroje – Část 6: Gravimetrická metoda zkoušení*. Český normalizační institut, 2003.

[2] ISO: *ISO-TR 20461-2000: Determination of uncertainty for volume measurements made using the gravimetric method*. International Organization for Standardization, 2000.

[3] Sartorius AG: *Operating Instructions Sartorius CP|Gemplus Series*, Sartorius AG, Germany, 2008, Publication No.:WCP6007-e08107.

[4] Druck Limited: *User Manual, DPI 140 Series Precision Pressure Indocator K023*, Druck Limited, Great Britain, 1995.

[5] GREISINGER electronic GmbH: *Instruction Manual, Operating Manual Precision Thermometer GMH3710 from Version 1.6,* GREISINGER electronic GmbH, Germany.

[6] COMET SYSTEM, s.r.o.: *Návod k použití, COMMETER D3121*, COMET SYSTEM, s.r.o., Česká republika.

[7] Horowitz, P., Hill, W.: *The Art of Electronics – 2nd ed.*, Cambridge, Cambridge University Press, 1989, ISBN-10: 0-521-37095-7.

[8] Tišnovský, P.: ROOT: *Seriál: Co se děje v počítači, díl Sériový port RS-232C* [online]. 27.11.2008 [cit. 10.5.2010]. Dostupný z WWW: <http://www.root.cz/clanky/seriovy-port-rs-232c/>

[9] Tumanski, S.: *Principes of Electrical Measurement*, Warsaw, CRC Press, 2006, ISBN-10: 0-7503-1038-3.

[10] Webster, J. G.: *Measurement, Instrumentation, and Sensors, The Handbook*, USA, CRC Press, 1999, ISBN-10: 0-8493-8347-1.

[11] Axelson, J.: *Serial Port Complete, Programming and Circuits for RS-232 and RS-485 Links and Networks*, USA, Lakeview Research, 2000, ISBN: 0-9650819-7-4.

[12] Labrosse, J. J.: *Embedded Systems Building Blocks*, Second Edition, USA, R&D Books, 2000, ISBN: 0-87930-604-1.

[14] Sartorius AG: *Syntax descripion, Sartorius SICS Interface*, Sartorius AG, Germany, 2010.

[15] Mettler-Toledo AG: *Reference Manual, METTLER TOLEDO Standard Interface Command Set for Excellence and Excellence Plus Balances*, Mettler-Toledo AG, Germany, 2007.

[16] COMET SYSTEM s.r.o.: *Kompaktní snímače teploty, vlhkosti, atmosférického tlaku s připojením na Ethernet* [online]. [cit. 13.4.2011]. Dostupný z WWW: <http://www.cometsystem.cz/prevodniky-teploty-seriove.htm>

[17] SCPI Consortium: *Standard Commands for Programmable Instruments (SCPI)* [online]. May 1999 [cit. 13.4.2011]. Dostupný z WWW: <http://www.ivifoundation.org/docs/SCPI-99.PDF/>

[18] Tišnovský, P.: ROOT: *Seriál: Co se děje v počítači, díl Universální sériová sběrnice USB* [online]. 15.1.2009 [cit. 13.3.2011]. Dostupný z WWW: <http://www.root.cz/clanky/universalni-seriova-sbernice-usb/>

[19] Tišnovský, P.: ROOT: *Seriál: Co se děje v počítači, díl Komunikační protokol universální sériové sběrnice* [online]. 22.1.2009 [cit. 13.3.2011]. Dostupný z WWW: <http://www.root.cz/clanky/komunikacni-protokol-universalni-seriove-sbernice/>

[20] Tišnovský, P.: ROOT: *Seriál: Co se děje v počítači, díl Přenos dat po universální sériové sběrnici* [online]. 29.1.2009 [cit. 13.3.2011]. Dostupný z WWW: <http://www.root.cz/clanky/prenos-dat-po-universalni-seriove-sbernici/>

[21] Axelson, J.: *USB Complete: The Developer's Guide (Complete Guides series) 4th edition*, USA, Lakeview Research, 2009, ISBN: 9781931448086.

[22] Peterson, L. L., Davie, B. S. : *Computer Networks: A Systems Approach 3rd edition*, USA, Morgan Kaufmann, 2003, ISBN: 9781558608320.

[23] Mathivanan, M. : *PC-based Instrumentation: Concepts and Practice*, India, Prentice-Hall of India Pvt.Ltd, 2007, ISBN: 8120330765.

[24] Mackay, S., Wright, E., *Reynders, D.: Practical industrial data networks: design, installation and troubleshooting*, USA, Newnes, 2004, ISBN: 9780750658072.

[25] Modbus Organization: *Modbus Application Protocol Specification V1.1b[online].December*, 2006 [cit. 23.4.2011]. Dostupný z WWW: <http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf>

[26] Modbus Organization: *Modbus over serial line specification and implementation guide V1.02*[online]. December, 2006 [cit. 23.4.2011]. Dostupný z WWW: <http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf>

[27] COMET SYSTEM s.r.o.: *Návod k použití, Programovatelný snímač teploty, relativní vlhkosti a dalších vlhkostních veličin T3311, T3313, T3411, vlhkosti, atmosférického tlaku se sériovým výstupem RS232, RS485* [online]. [cit. 29.4.2011]. Dostupný z WWW: <http://www.cometsystem.cz/manuals/i-snc-t33_4_11+t73_4_10.pdf>

[28] Arlow, J., Neustadt, I.: *UML 2 a unifikovaný proces vývoje aplikací*, 2. vydání, Czech Republic, CPress, 2004, ISBN: 978-80-251-1503-9.

[29] Bishop, J.: *C# návrhové vzory,* 2. vydání, Czech Republic, Zoner Press, 2010, ISBN 978-80-7413-076-2

[30] Gamma, E., Helm, R., Johnson, R., Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software,*1st edition, USA, Addison-Wesley Professional, 1994, ISBN 978-0201633610

[31] ÚNMZ: TNI 01 0115*: Mezinárodní metrologický slovník - Základní a všeobecné pojmy a přidružené termíny (VIM),* Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2009.

[32] ISO: *ČSN EN ISO 8865-1:2003: Pístové titrační přístroje – Část 1: Část 1: Termíny, všeobecné požadavky a doporučení pro uživatele*. Český normalizační institut, 2003.

Seznam příloh

Seznam zkratek

ADU - Application Data Unit

Bd - baud

CRC - Cyclical Redundancy Checking

ČMI - Český metrologický institut

DCE - Data Communication Equipment

DTE - Data Terminal Equipment

EIA - Electronics Industries Association

GLP - Good Laboratory Practice

GUI – Graphic User Interface

LRC - Longitudinal redundancy check

Modbus RTU – Modbus Remote Terminal Unit

MT-SICS - METTLER TOLEDO - Standard Interface Command Set

NRZI - Non Return To Zero Inverted

PDU – Protocol Data Unit

PLC - Programmable Logic Controllers

RI - Ring Indicator

RS - Recommended Standard

SDU – Serial Data Unit

SICS - *Standard Interface Common Set*

TMDS - Transition Minimized Differential Signaling

UART -Universal asynchronous receiver/transmitter

Příloha č. 2 - Postup při výpočtu Modbus CRC

”

1. Naplnit 16bitový registr hodnotou 0xFFFF (všechny bity nastaveny na 1). Nazvěme tento registr „CRC registrem“.
2. Provést logickou funkci Exclusive OR s prvním osmi bitovým Byte zprávy se spodními osmi bity CRC registru, výsledek uložit do CRC registru.
3. Posunout obsah CRC registru o jeden bit vpravo (směrem k LSB), jako horní bit CRC registru vložit 0. Zapamatovat si hodnotu nejnižšího odsouvaného bitu (LSB).
4. Pokud LSB byl 0, pak opakujte krok 3 (další posuv). Pokud LSB byl 1, pak provést Exclusive OR CRC registru s hodnotou 0xA001.
5. Opakovat krok 3 a 4 dokud neproběhne osm posuvů. Po osmi posuvech je osmi bitový Byte zpracován.
6. Opakovat kroky 2 až 5 na další osmi bitové Byte zprávy dokud nebudou všechny Byte zpracovány.
7. Na konec po zpracování všech Byte zprávy je v CRC registru uložena hodnota kontrolního součtu.
8. Při připojování kontrolního součtu ke zprávě se jako první vysílá dolní Byte CRC registru, potom horní Byte CRC registru.“, převzato z [27].

Příloha č. 3 - Vzorek analyzovaných dat z teploměru GREISINGER GMH3710

Port opened by process "GMHKonfig.exe" (PID: 2952)

FE C0 73 ?As

Answer: 27.12.2009 20:39:09.65264 (+0.0701 seconds)

FE C5 68 CD 40 3C FC 09 ?AhÍ@<ü.
28 (

Request: 27.12.2009 20:39:10.92364 (+0.2704 seconds)

FE F2 ED 2F 00 92 ?oí/.?

Answer: 27.12.2009 20:39:10.98364 (+0.0601 seconds)

FE F5 F8 2F 00 92 FF 01 ?oo/.?y.
2F /

Request: 27.12.2009 20:39:10.22364 (+0.2403 seconds)

FE F2 ED 01 00 EA ?oí..e

Answer: 27.12.2009 20:39:10.26364 (+0.0401 seconds)

FE F5 F8 01 00 EA 24 12 ?oo..e$.
7B {

Request: 27.12.2009 20:39:10.45464 (+0.1903 seconds)

FE 00 3D ?.=

Answer: 27.12.2009 20:39:10.55464 (+0.1001 seconds)

FE 05 26 71 00 48 F3 74 ?.&q.Hót
9F ?

Request: 27.12.2009 20:39:10.78464 (+0.2303 seconds)

FE F2 ED 37 00 6D ?oí7.m

Answer: 27.12.2009 20:39:10.81464 (+0.0300 seconds)

FE F7 F6 37 01 6A 7A FF ?÷ö7.jzy
2C 07 30 04 ,.0.

Request: 27.12.2009 20:39:11.04464 (+0.2203 seconds)

FE F2 ED 36 00 78 ?oí6.x

Answer: 27.12.2009 20:39:11.14564 (+0.1001 seconds)

FE F7 F6 36 01 7F 79 00 ?÷ö6. y.
E0 DE 34 1F a?4.

Request: 27.12.2009 20:39:11.27564 (+0.1302 seconds)

FE F2 ED 35 00 47 ?oí5.G

Answer: 27.12.2009 20:39:11.32564 (+0.0501 seconds)

FE F5 F8 35 00 47 FF 01 ?oo5.Gy.
2F /

Request: 27.12.2009 20:39:11.35564 (+0.0300 seconds)

FE F2 ED 38 00 AE ?oí8.Ź

Answer: 27.12.2009 20:39:11.39564 (+0.0401 seconds)

FE F5 F8 38 00 AE FF 01 ?oo8.Źy.
2F /

Request: 27.12.2009 20:39:11.44564 (+0.0501 seconds)

FE C0 73 ?As

Answer: 27.12.2009 20:39:11.48564 (+0.0401 seconds)

FE C5 68 CD 40 3C FC 09 ?AhÍ@<ü.
28 (

Request: 27.12.2009 20:39:11.50564 (+0.0200 seconds)

FE F2 ED 38 00 AE ?oí8.Ź

Answer: 27.12.2009 20:39:11.54564 (+0.0401 seconds)

FE F5 F8 38 00 AE FF 01 ?oo8.Źy.
2F /

Request: 27.12.2009 20:39:11.57564 (+0.0300 seconds)

FE F2 ED 35 00 47 ?oí5.G

Answer: 27.12.2009 20:39:11.61564 (+0.0401 seconds)

FE F5 F8 35 00 47 FF 01 ?oo5.Gy.
2F /

Request: 27.12.2009 20:39:11.63564 (+0.0200 seconds)

FE 00 3D ?.=

Answer: 27.12.2009 20:39:11.66564 (+0.0300 seconds)

FE 05 26 71 00 48 F3 73 ?.&q.Hós
8A ?

Request: 27.12.2009 20:39:11.69564 (+0.0200 seconds)

FE F4 FF DF 00 86 FF 32 ?ôyß.?y2
B6 ť

Answer: 27.12.2009 20:39:11.84664 (+0.1502 seconds)

FE 51 8D ?Q?

Request: 27.12.2009 20:39:12.93664 (+0.0901 seconds)

FE F2 ED 1B 00 3F ?oí..?

Answer: 27.12.2009 20:39:12.08664 (+0.1502 seconds)

FE 51 8D ?Q?

Request: 27.12.2009 20:39:12.10664 (+0.0200 seconds)

FE F2 ED 16 00 D6 ?oí..Ö

Answer: 27.12.2009 20:39:12.26664 (+0.1602 seconds)

FE 51 8D ?Q?

Request: 27.12.2009 20:39:13.06764 (+0.8012 seconds)

FE F2 ED 1B 00 3F ?oí..?

Příloha č. 4 - Tabulka porovnání typů přenosů u USB

Table 2-1: Each of the USB's four transfer types is suited for different uses.

Transfer Type	Control	Bulk	Interrupt	Isochronous
Typical Use	Identification and configuration	Printer, scanner, drive	Mouse, keyboard	Streaming audio, video
Support required?	yes	no	no	no
Low speed allowed?	yes	no	yes	no
Maximum packet size; maximum guaranteed packets/interval (SuperSpeed).	512; none	1024; none	1024; 3 / 125 µs	1024; 48 / 125 µs
Maximum packet size; maximum guaranteed packets/interval (high speed).	64; none	512; none	1024; 3 / 125 µs	1024; 3 / 125 µs
Maximum packet size; maximum guaranteed packets/interval (full speed).	64; none	64; none	64: 1 / ms	1023; 1 / ms
Maximum packet size; maximum guaranteed packets/interval (low speed).	8; none	not allowed	8; 1 / 10 ms	not allowed
Direction of data flow	IN and OUT	IN or OUT	IN or OUT (IN only for USB 1.0)	IN or OUT
Reserved bandwidth for all transfers of the type	10% at low/full speed, 20% at high speed & SuperSpeed	none	90% at low/full speed, 80% at high speed and SuperSpeed (isochronous and interrupt combined, maximum)	
Message or Stream data?	message	stream	stream	stream
Error correction?	yes	yes	yes	no
Guaranteed delivery rate?	no	no	no	yes
Guaranteed latency (maximum time between transfers attempts)?	no	no	yes	yes

Příloha č.4: Tabulka porovnání typů přenosů u USB. Převzato z [21]

Příloha č. 5 - Parametry rozhraní elektromechanické váhy Sartorius CPA225D-OCE

Rychlosti: 150, 300, 600, 1200, 2400, 4800, 9600 a 19200 baudů (1200)

Parity: lichá, sudá, space, mark (lichá)

Formát: 1start bit, 7bitu ASCII a 1 nebo 2 stopbity (1stop bit)

Handshake: softwarový (XON/XOFF)

hardwarový (CTS/DTR) - (tovární nastavení – dva znaky po CTS)

Operační mód: SBI(Standard Sartorius interface) - (tovární nastavení)

Formát výstupních dat: 16 nebo 22znaků

Příloha č. 6 - Parametry rozhraní přesného barometru Druck DPI140

Rychlosti: 150, 300, 600, 1200, 2400, 4800 a 9600 baudů (1200)

Parity: není

Formát: 1start bit, 8 datových bitů (MSB je 0) a 1 stop bit

Handshake: softwarový (XON/XOFF)

hardwarový (CTS/DTR) - (tovární nastavení)

Módy výstupních dat: tiskárna, počítač

Příloha č. 7 - Parametry USB převodníku USB3100

Převodník využívá CP210x firmy Silicon Labs s parametry:

Rychlosti: 300, 600, 1200, 2400, 4800 až 921600 baudů

Parity: žádná, sudá, lichá

Formát: 8 datových bitů a 1 stop bit

Příloha č. 8 – Comet T3311 - Modbus registry zařízení

Proměnná	Jednotka	Adresa[hex][x]	Adresa[dec][x]	Formát	Velikost	Status
Měřená teplota	[°C] [°F]*	0x0031	49	Int*10	BIN16	R
Adresa zařízení	[-]	0x2001	8193	Int	BIN16	R/W**
Kód přenosové rychlosti	[-]	0x2002	8194	Int	BIN16	R/W**
Sériové číslo zařízení Hi	[-]	0x1035	4149	BCD	BIN16	R
Sériové číslo zařízení Lo	[-]	0x1036	4150	BCD	BIN16	R
Verze Firmware Hi	[-]	0x3001	12289	BCD	BIN16	R
Verze Firmware Lo	[-]	0x3002	12290	BCD	BIN16	R
Měřená relativní vlhkost	[%]	0x0032	50	Int*10	BIN16	R
Hodnota počítané veličiny*		0x0033	51	Int*10	BIN16	R

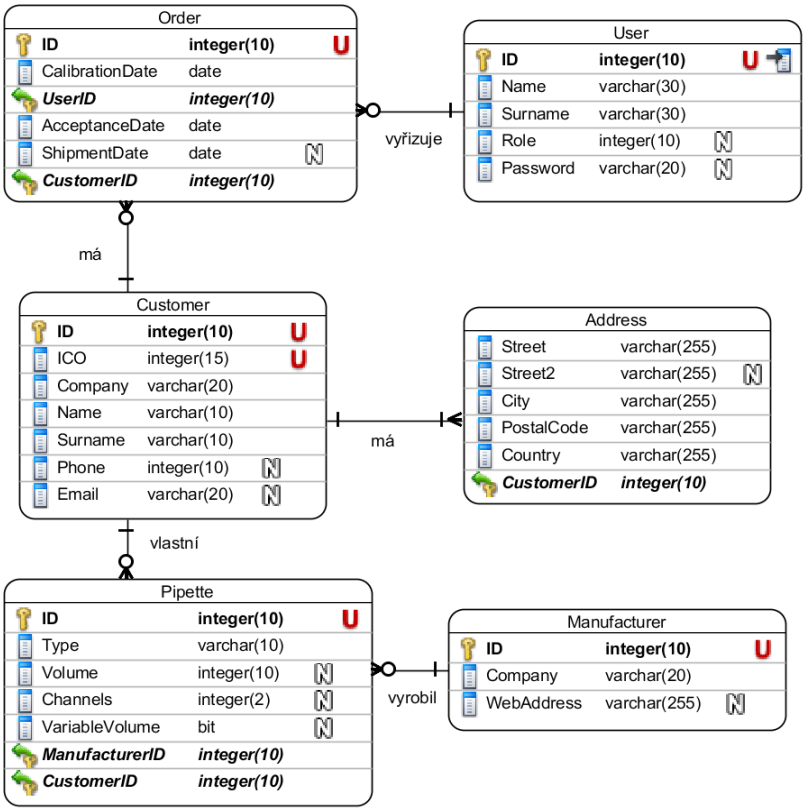
Tabulka č. : Comet T3311 – Modbus registry zařízení, převzato z [27]

„**Vysvětlivky:**

- podle typu a nastavení snímače (pomocí uživatelského software)
- Int*10 registr je ve formátu integer*10
- R registr je určen jen pro čtení
- W** registr je určen pro zápis, podrobněji viz popis funkce 16 (0x10): Nastavení více 16 bitových registrů (Write Multiple Registers)
- [x] Při přenosu jsou adresy registrů indexovány od nuly, tj. registr 0x31 se fyzicky po sběrnici vyšle jako hodnota 0x30, 0x32 jako 0x31…)

Pozn. V případě potřeby vyčítat měřené hodnoty ze zařízení s větším rozlišením než jedna desetina, jsou měřené hodnoty ve zařízení uloženy i ve „Float“ formátu, který ovšem není přímo kompatibilní s IEEE754“ převzato z [27].

Příloha č. 9 - Konceptuální návrh databáze



Příloha č. 10 Konceptuální návrh